Part I

Foundations

Part II

Bargaining

Part III

Auctions and Mechanism Design

Part IV

Matching

Chapter 6

Matching Theory

6.1 The Housing Market: Allocation with Endowments

We now move from settings where monetary transfers are available (auctions, mechanisms with payments) to *allocation problems without money*. No taxes, no subsidies, no side payments—agents simply trade indivisible objects among themselves. This is the classic *housing market* model introduced by Shapley and Scarf (1974).

6.1.1 Housing Market and the Core

Definition 6.1: Housing Market

A *housing market* is a triple $(\mathcal{P}, H, \succ)$ where:

- $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ is a finite set of *agents* ("people");
- $H = \{h_1, h_2, \ldots, h_n\}$ is a finite set of *houses* ("offices"), with $|H| = |\mathcal{P}|$;
- Each agent P_i is initially endowed with house h_i and has a strict preference ordering $\succ_i$ over H.

An *allocation* is a bijection $Q : \mathcal{P} \to H$. That is, Q assigns exactly one house to each agent.

The key question is: can we find an allocation that is *stable* in the sense that no group of agents can profitably rearrange houses among themselves?

Definition 6.2: Core

An allocation Q^* is in the core if there is no coalition S and no allocation Q' of $\{h_i : P_i \in S\}$ to S such that $Q'(P_i) \succsim_i Q^*(P_i)$ for all $P_i \in S$ and $Q'(P_j) \succ_j Q^*(P_j)$ for some $P_j \in S$.

Intuitively, Q^* is in the *core* if there is no subset $S \subseteq \mathcal{P}$ (a *blocking coalition*) such that the members of S can reallocate their initial endowments $\{h_i : P_i \in S\}$ among themselves

in a way that makes every member of S weakly better off, and at least one member strictly better off.

Remark (Why "Core" and Not Just Pareto Efficiency?).

> Pareto efficiency only rules out improvements where *everyone* can be made better off (the grand coalition $S = \mathcal{P}$). The core is a much stronger requirement: it rules out deviations by *any* subset of agents, including small coalitions. In particular, individual rationality (no agent envies their own initial endowment) is a special case of the core condition with $|S| = 1$.

6.1.2 Top Trading Cycle (TTC)

The *Top Trading Cycle* algorithm, attributed to David Gale by Shapley and Scarf (1974), provides an elegant procedure to find a core allocation.

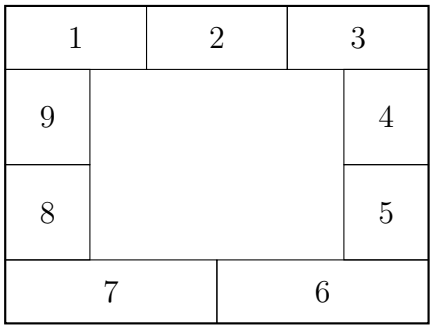
Top Trading Cycle Algorithm

Given a housing market $(\mathcal{P}, H, \succ)$ with initial endowment $P_i \mapsto h_i$:

1. **Point:** Each remaining agent points to the agent who currently owns their most preferred house (among remaining houses).
2. **Trade:** Since the number of agents is finite and each agent points to exactly one other agent, at least one cycle must exist. Identify all cycles. Each agent in a cycle receives the house they pointed to (i.e., the house owned by the agent they pointed to).
3. **Remove:** Remove all agents and houses involved in cycles from the market.
4. **Repeat:** Return to Step 1 with the reduced market. Terminate when all agents have been assigned a house.

Example (The Office Problem).

Consider 9 people $\mathcal{P} = \{P_1, \ldots, P_9\}$ and 9 offices $H = \{1, \ldots, 9\}$, arranged around a corridor as follows:



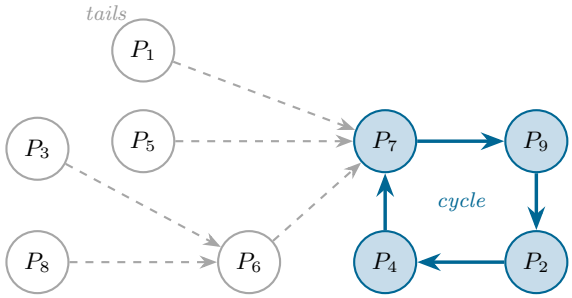
Person P_i initially occupies office i. The agents' strict preference orderings (listed from most preferred to least preferred) are:

Rank	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
1	7	4	6	7	7	7	9	6	2
2	6	9	2	1	1	1	3	2	5
3	9	6	8	5	8	2	6	4	4
4	4	2	1	8	4	4	2	3	8
5	8	5	7	9	3	5	5	7	3
6	5	3	4	3	5	3	4	5	1
7	1	7	5	2	6	8	8	1	9
8	2	8	9	6	9	9	7	8	6
9	3	1	3	4	2	6	1	9	7

We apply TTC to the office problem defined above. The logic of each round is as follows:

- Every remaining agent *points to* the current owner of their most-preferred available office, forming a directed graph in which each node has out-degree exactly one.
- Following the arrows traces directed **chains**. Because each node has exactly one outgoing edge, every chain must eventually revisit a node—so every chain either *closes into a cycle* or *feeds into* a cycle found by another chain.
- **Cycle members** trade immediately (each receives the office of the agent they pointed to). **Tail agents** (those whose chains lead into the cycle without closing) are held over for the next round.

Round 1. All nine agents with offices $\{1, \ldots, 9\}$.



Chains and cycle detection.

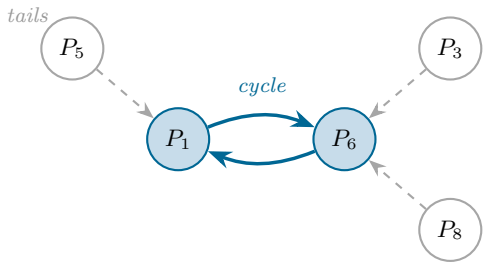
- Start from P_1: $P_1 \to P_7 \to P_9 \to P_2 \to P_4 \to P_7$ — the chain revisits P_7. **Cycle found:** $P_7 \to P_9 \to P_2 \to P_4 \to P_7$.
- P_1 (arriving at P_7), as well as P_5 (pointing directly to P_7), P_6 (pointing to P_7), P_3 and P_8 (pointing to P_6) are all tails. They do not close any new cycle.

Cycle trade:

$$P_7 \leftarrow \text{office } 9, \quad P_9 \leftarrow \text{office } 2, \quad P_2 \leftarrow \text{office } 4, \quad P_4 \leftarrow \text{office } 7.$$

Remove $\{P_2, P_4, P_7, P_9\}$ and offices $\{2, 4, 7, 9\}$. Tails $\{P_1, P_3, P_5, P_6, P_8\}$ proceed.

Round 2. Remaining agents: $\{P_1, P_3, P_5, P_6, P_8\}$; available offices: $\{1, 3, 5, 6, 8\}$.

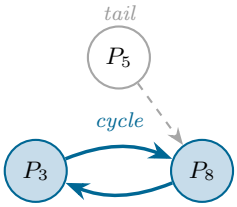


Chains and cycle detection.

- Start from P_5: $P_5 \to P_1 \to P_6 \to P_1$ — revisits P_1. **Cycle found:** $P_1 \leftrightarrow P_6$.
- P_3 and P_8 both point to P_6 (into the cycle), and P_5 points to P_1 (into the cycle). All three are tails.

Cycle trade: $P_1 \leftarrow$ office 6, $\quad P_6 \leftarrow$ office 1.
Remove $\{P_1, P_6\}$ and offices $\{1, 6\}$. Tails $\{P_3, P_5, P_8\}$ proceed.

Round 3. Remaining: $\{P_3, P_5, P_8\}$; available offices: $\{3, 5, 8\}$.



Chains and cycle detection.

- $P_3 \to P_8 \to P_3$ — revisits P_3. **Cycle found:** $P_3 \leftrightarrow P_8$.
- $P_5 \to P_8$ feeds into the cycle; P_5 is a tail.

Cycle trade: $P_3 \leftarrow$ office 8, $\quad P_8 \leftarrow$ office 3.
Remove $\{P_3, P_8\}$ and offices $\{3, 8\}$.

Round 4. Only P_5 remains with office 5. With a single agent, the "chain" is a trivial self-loop: $P_5 \to P_5$. The self-loop is itself a cycle, so P_5 keeps office 5.

The final TTC allocation is:

Agent	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Office	6	4	8	7	5	1	9	3	2

Remark (Cycle Structure in TTC).

In each round, the pointing graph is a *functional graph*: every node has out-degree exactly one. A fundamental property of functional graphs is that each weakly connected component contains *exactly one* directed cycle, with all remaining nodes forming tails that feed into it. Consequently:

- **Existence:** At least one cycle is guaranteed to exist in every round, since the finite graph must eventually revisit a node along any directed path.
- **Multiplicity:** The number of cycles per round equals the number of weakly connected components—it may be one or many, depending on the preference profile. Uniqueness of cycles *per round* does not hold in general.
- **Uniqueness within a chain:** Following the arrows from any fixed starting node traces a single directed path, which must eventually close into *exactly one* cycle. This per-chain uniqueness is what makes TTC well-defined: each agent belongs to at most one cycle and receives a unique assignment in the round that cycle is identified.

6.1.3 Properties of TTC

TTC Implements the Core

Theorem 6.3: TTC Allocation is in the Core

The allocation produced by the TTC algorithm is in the core of the housing market. Moreover, when preferences are strict, the core contains a unique allocation, and TTC finds it.

Proof for Theorem

The intuition is as follows. Agents removed in Round 1 each obtain their overall top choice—no coalition including any of them can make them strictly better off, since they already have their first-best house. Once Round 1 agents are settled, Round 2 agents obtain their best feasible house given Round 1 assignments, and so on. Any blocking coalition would need to offer some agent a house better than what TTC gave them, but that house has already been claimed by someone removed in an earlier round who would never agree to give it up.

We proceed by induction on the round of removal. For each round r, let C_r denote the set of agents removed in round r (i.e., the union of all cycles identified and executed in that round), and let $H_r = \{\mu(i) : i \in C_r\}$ denote the corresponding set of houses allocated in round r.

Base case (Round 1). Every agent removed in Round 1 belongs to a cycle in which each member points to the owner of their globally top-ranked house. Each such agent therefore receives their first-best house. No coalition S can block this allocation: to improve upon the outcome for any Round-1 agent $i \in S$, the coalition would need to reassign i's own top choice to i—but i already has it.

Inductive step. Suppose that for all rounds $r' < r$, the agents removed in rounds $1, \ldots, r' - 1$ receive their best house within the set of houses held by agents in those same rounds, and no coalition can improve upon their assignment using only houses from $\bigcup_{r'' \leq r'} H_{r''}$. Consider agents removed in round r. Each such agent receives their most preferred house among all houses still available—i.e., not yet claimed by any agent removed in rounds $1, \ldots, r - 1$.

Suppose for contradiction that some blocking coalition S exists. Let $i \in S$ be an agent who is made strictly better off under the proposed reassignment, and let h' be the house S proposes to give i, where $h' \succ_i \mu(i)$. Since $\mu(i)$ is already i's best available house in round r, the house h' must have been removed in some earlier round $r' < r$—it belongs to some agent $j \in C_{r'}$ who exited in round r' with $\mu(j) = h'$. For S to reassign h' to i, it must include j. But by the inductive hypothesis, $\mu(j) = h'$ is j's best house within $H_{r'}$; giving up h' makes j strictly worse off, so j would not participate in S. Contradiction.

Hence the TTC allocation is in the core. For uniqueness under strict preferences: any core allocation must give Round-1 agents their top choice (otherwise the coalition C_1 itself is a blocking coalition), and by induction any core allocation must coincide with TTC round by round. Thus the core is a singleton, and TTC finds it.

Corollary 6.4: TTC is Individually Rational

The allocation produced by TTC is individually rational: every agent weakly prefers their assigned house to their own initial endowment. That is, $\mu(i) \succsim_i h_i$ for all $i \in \mathcal{P}$.

Proof for Corollary.

This follows immediately from the core property. Suppose for contradiction that some agent i strictly prefers their endowment to their TTC allocation: $h_i \succ_i \mu(i)$. Then the singleton coalition $S = \{i\}$, using the trivial allocation $Q'(i) = h_i$ (agent i keeps their own house), satisfies $Q'(i) \succ_i \mu(i)$. This means S blocks μ, contradicting the fact that μ is in the core. Hence no such i can exist.

TTC is Order Independent

Theorem 6.5: TTC is Order Independent (Strict Preferences)

Suppose all agents have strict preferences over objects. When there are multiple cycles in a given round, the TTC outcome does not depend on the order in which cycles are identified and removed. That is, under strict preferences, the final allocation is unique regardless of which cycle is processed first.

Proof for Theorem

The proof reduces to one key structural observation about functional graphs.

Claim

Under strict preferences, in any round of TTC, all cycles in the pointing graph are *vertex-disjoint*.

Proof for Claim.

Under strict preferences, each agent has a unique top-ranked available object, so the pointing graph is a functional graph: every node has out-degree exactly one. Suppose for contradiction that two distinct cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ share a node v. Then v would need two distinct outgoing edges—one continuing around $\mathcal{C}_1$ and one around $\mathcal{C}_2$—contradicting out-degree one. Hence all cycles are vertex-disjoint.

Since any two coexisting cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ are vertex-disjoint, the agents and houses in $\mathcal{C}_1$ are entirely distinct from those in $\mathcal{C}_2$. Removing $\mathcal{C}_1$ first therefore does not affect the membership or the pointing structure of $\mathcal{C}_2$, and vice versa. More precisely:

- The agents in $\mathcal{C}_2$ are still present after $\mathcal{C}_1$ is removed, and their preferences over the remaining houses are unchanged.
- The agents in $\mathcal{C}_2$ still point to the same targets they did before (none of their targets were in $\mathcal{C}_1$), so $\mathcal{C}_2$ remains a valid cycle in the reduced graph.

By symmetry, removing $\mathcal{C}_2$ first leaves $\mathcal{C}_1$ intact. In either order, both cycles are eventually removed and their members receive exactly the same houses. Since the argument applies to every pair of coexisting cycles in every round, the final allocation is independent of the order in which cycles are processed.

Remark (Order Dependence Under Weak Preferences).

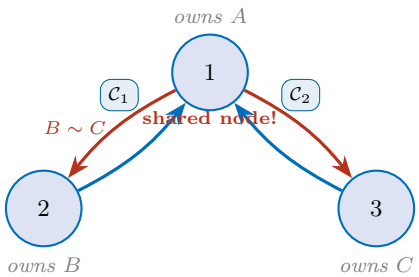
The strictness assumption embedded in our definition of a housing market is not merely a technical convenience—it is essential for TTC to be well-defined. To see what goes wrong if we relax it, suppose we allow agents to be *indifferent* between objects (i.e., $\succ_i$ is replaced by a weak order $\succsim_i$). Then an agent with multiple top-ranked objects has out-degree greater than one in the pointing graph, vertex-disjointness of cycles can fail, and different cycle-processing orders may yield different allocations.

Counterexample. Consider three agents with three houses:

$$\text{agent 1 owns } A, \quad \text{agent 2 owns } B, \quad \text{agent 3 owns } C,$$

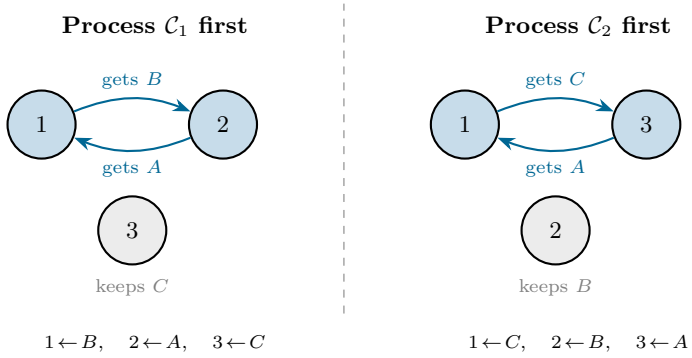
with weak preferences $1 : B \sim C \succ A, \quad 2 : A \succ B \succ C, \quad 3 : A \succ C \succ B.$

Pointing graph. Agent 1 is indifferent between B and C and simultaneously points to both agents 2 and 3. Agents 2 and 3 each point to agent 1. The resulting graph has out-degree 2 at node 1, and contains two cycles sharing that node:



The two cycles $\mathcal{C}_1 = (1 \to 2 \to 1)$ and $\mathcal{C}_2 = (1 \to 3 \to 1)$ are *not* vertex-disjoint. Processing

them in different orders gives different outcomes:



The two orders produce distinct allocations, confirming that TTC is not well-defined under weak preferences without an additional tie-breaking rule.

The standard remedy is to impose an exogenous **tie-breaking rule** (e.g., a lottery over objects) before running TTC, converting weak preferences into strict ones. This restores out-degree one, vertex-disjoint cycles, and a well-defined outcome. However, the final allocation then depends on the tie-breaking rule chosen, so TTC under weak preferences is best understood as a *family* of mechanisms indexed by tie-breaking rules, rather than a single deterministic procedure.

TTC is Strategy-Proof

Theorem 6.6: TTC is Individual Strategy-Proof

Under the TTC mechanism, truthful reporting of preferences is a dominant strategy for every agent. That is, regardless of what other agents report, no agent can obtain a strictly better outcome by misreporting their preferences.

Proof for Theorem

Fix any agent i and any reports P_{-i} by all other agents. Let μ denote the TTC outcome under truthful reporting, with $\mu(i) = o^*$, and let r be the round in which i exits under truthful TTC. We show that no misreport P_i' yields i a strictly better object.

Key observation. An agent's report determines only *whom they point to*; it does not affect whom others point to. Since all $j \neq i$ report truthfully, the pointing behavior of every $j \neq i$ depends only on j's own preferences and the current available set—both of which are unaffected by i's report.

We present two proofs, each illuminating a different aspect of the mechanism.

Proof 1 (by contradiction on the object obtained). Suppose i misreports and obtains $o' \succ_i o^*$. We derive a contradiction in two cases.

- **Case 1: o' was taken in round $r' < r$ under truth-telling.** For i to obtain o', i must be part of the cycle containing o' in round r' under the misreport. This requires some agent $j \neq i$ to point to i in round r', closing the cycle through i. But j's pointing

depends only on j's preferences and the available set at round r'. As long as i is still present at round r', the agents and objects removed in rounds $1, \ldots, r'-1$ are exactly those from cycles not involving i—the same as under truth-telling (since those cycles form independently of i's report). Hence the available set at round r' is identical under both runs, and so is j's pointing. Since no j points to i in round r' under truth-telling (no cycle involving i forms before round r), no j does so under misreport either. The cycle closes without i. Contradiction.

- **Case 2: o' is available in round r under truth-telling.** Then $o' \in A_r$, the available set when i's truthful cycle forms. But under truth-telling i points to the owner of their top choice in A_r, which would be o' if $o' \succ_i o^*$—contradicting $\mu(i) = o^*$.

Proof 2 (by analyzing the exit round). Suppose i misreports and exits in round $r' \neq r$. We show neither case improves i's outcome.

- **Case 1: $r' > r$ (i exits later).** In TTC, each agent exits in the round their cycle forms, receiving their most preferred object among those still available *at that round*. By construction, $A_{r'} \subseteq A_r$: the longer i stays, the more objects get claimed by earlier cycles. Hence i's best available object weakly worsens as the round increases, and exiting later cannot improve i's outcome.

- **Case 2: $r' < r$ (i exits earlier).** For i to exit in round $r' < r$, i must join a cycle in that round. This requires some $j \neq i$ to point toward i (to close the cycle through i). Under truth-telling, j points to i in round r' if and only if h_i is j's top available object at round r'. Since the available set at round r' is the same under both runs (cycles before round r' do not involve i and form identically), j already points to i in round r' under truth-telling. But then j never exits before round r', so j is still present in round r under truth-telling. This means $h_j \in A_r$, the available set when i exits truthfully. Since $\mu(i) = o^*$ is i's top choice from A_r and $h_j \in A_r$, we have $o^* \succsim_i h_j$. Joining j's cycle early to obtain h_j cannot improve upon o^*.

In all cases, no misreport yields i a strictly better object than o^*. Since i and P_{-i} were arbitrary, truth-telling is a dominant strategy for every agent. ∎

The two proofs above capture complementary intuitions about why manipulation fails in TTC.

- **Proof 1** asks: *can i obtain a specific better object o'?* It shows the answer is no by ruling out each possible location of o': either o' was already taken (and i cannot intercept its cycle, since no one points back to i), or o' was available all along (and i would have gotten it honestly).

- **Proof 2** asks: *can i benefit by changing which round it exits?* It shows the answer is no in both directions. Exiting later only shrinks the available set. Exiting earlier forces i to join a chain that was already pointing toward i under truth-telling—meaning the object at the end of that chain was already in i's truthful available set A_r, and therefore no better than o^*.

At the deepest level, both proofs rest on the same asymmetry: *i controls only whom it*

points to, not who points back. The objects reachable by i are ultimately determined by who is willing to trade with i—and that depends on others' preferences, which no misreport can change.

A natural follow-up question is whether TTC is also *group strategy-proof*: can a coalition of agents coordinate their misreports to make every member of the coalition strictly better off? The answer is **NO**—TTC is group strategy-proof. No coalition can jointly misreport preferences and make all coalition members strictly better off. This is a stronger result that follows from the fact that TTC implements the unique strict core allocation, combined with the structure of the trading cycle procedure.

Theorem 6.7: TTC is Strictly Group Strategy-Proof

TTC is group strategy-proof: there is no coalition $S \subseteq \mathcal{P}$ and no coordinated misreport P'_S such that every member of S obtains a strictly better object than under truthful reporting.

Proof for Theorem

The proof has two parts that mirror—and extend—the individual strategy-proofness argument. Individual SP shows that no single agent can infiltrate a better cycle, because they cannot control who points back to them. Group SP faces a richer challenge: even if coalition members coordinate their pointing, they still cannot control the pointing of agents outside S, and—crucially—the core uniqueness of TTC prevents them from profitably rearranging objects among themselves either.

Suppose for contradiction that S misreports P'_S and obtains allocation μ' with $\mu'(i) \succ_i \mu(i)$ for all $i \in S$, where μ is the truthful TTC outcome. Partition S's gains by the origin of the improved objects: outside S, and inside S (redistribution).

Claim

No S-member can gain an object initially owned by an agent outside S.

Proof for Claim.

Suppose $i \in S$ gains object $o' = \mu'(i) \succ_i \mu(i)$, where o' is initially owned by some $j \notin S$. Fix the reports of all other agents at $(P'_{S\setminus\{i\}}, P_{-S})$ and apply the individual strategy-proofness argument to i alone. Regardless of what others report, i cannot benefit from misreporting; in particular, the same two-case argument applies verbatim:

- If o' was removed in an earlier round under this fixed profile, $j \notin S$ reports truthfully, so j's pointing behavior depends only on j's own preferences and the available set—neither of which i's report can affect. No agent in the cycle containing o' points back to i, so i cannot join that cycle regardless of i's misreport. Contradiction.
- If o' is still available in i's exit round, then $o' \succ_i \mu(i)$ implies i would have pointed to j (owner of o') under truthful reporting and obtained o' honestly— contradicting $\mu(i) \neq o'$.

Hence every $i \in S$ can only gain objects initially endowed to other S-members. That is, $\{\mu'(i) : i \in S\} \subseteq \{h_i : i \in S\}$.

Claim

S cannot profitably redistribute its own endowments.

Proof for Claim.

By the previous claim, the objects received by S-members under μ' are drawn entirely from $\{h_i : i \in S\}$. Define an allocation ν for S by $\nu(i) = \mu'(i)$ for all $i \in S$. Then ν is a redistribution of $\{h_i : i \in S\}$ among S such that $\nu(i) \succ_i \mu(i)$ for all $i \in S$. This means S constitutes a *blocking coalition* for μ under true preferences, contradicting the fact that μ is in the strict core.

Both parts yield contradictions, so no coalition S can jointly misreport to make all its members strictly better off.

Theorem 6.8: TTC is Weakly Group Strategy-Proof

The TTC mechanism is weakly group strategy-proof: there is no coalition $S \subseteq \mathcal{P}$ and joint misreport P'_S such that the resulting outcome μ' satisfies

$$\mu'(i) \succsim_i \mu(i) \text{ for all } i \in S, \quad \text{with} \quad \mu'(i^*) \succ_{i^*} \mu(i^*) \text{ for some } i^* \in S.$$

Proof for Theorem

Suppose for contradiction that such S and P'_S exist. Since preferences are strict, every $i \in S$ either keeps exactly the same object ($\mu'(i) = \mu(i)$) or gets a strictly better one. Partition S accordingly:

$$S^+ = \{i \in S : \mu'(i) \succ_i \mu(i)\} \neq \emptyset, \qquad S^0 = \{i \in S : \mu'(i) = \mu(i)\}.$$

Run TTC under the misreport profile (P'_S, P_{-S}). Among all S^+-members, let r^* be the earliest round in which any S^+-member exits, and let $i^{**} \in S^+$ be such a member. Let $\mathcal{C}$ be the TTC cycle that i^{**} belongs to in round r^*.

In cycle $\mathcal{C}$, agents trade their initial endowments cyclically: writing $\mathcal{C} = (a_1 \to a_2 \to \cdots \to a_k \to a_1)$, each a_l gets $h_{a_{l+1}}$ (indices mod k). In particular, $\mu'(a_l) = h_{a_{l+1}}$ for each l.

Case 1: $\mathcal{C} \subseteq S$.

All members of $\mathcal{C}$ belong to S, so by the weak improvement assumption,

$$\mu'(a_l) = h_{a_{l+1}} \succsim_{a_l} \mu(a_l) \quad \text{for all } a_l \in \mathcal{C},$$

with $h_{a_{l^*+1}} \succ_{i^{**}} \mu(i^{**})$ for $i^{**} \in \mathcal{C}$. The coalition $\mathcal{C}$ can execute exactly this trade under their *true* preferences: each a_l contributes endowment h_{a_l} and receives $h_{a_{l+1}}$, a redistribution of $\{h_{a_l} : a_l \in \mathcal{C}\}$. Every member weakly benefits, and i^{**} strictly benefits. This means $\mathcal{C}$ is a blocking coalition for μ under true preferences, contradicting the fact that μ is in the strict core.

Case 2: $\mathcal{C} \not\subseteq S$.

$\mathcal{C}$ contains $i^{**} \in S^+$ and at least one agent $j_0 \notin S$. We consider two sub-cases based on whether the non-S members of $\mathcal{C}$ are harmed by the trade.

Sub-case 2a: $\mu'(j) = h_{n(j)} \succsim_j \mu(j)$ for every $j \in \mathcal{C} \setminus S$.

Then every agent in $\mathcal{C}$ weakly benefits from the cycle trade (under true preferences), and i^{**} strictly benefits. Again, $\mathcal{C}$ is a blocking coalition for μ, contradicting the strict core.

Sub-case 2b: Some $j_0 \in \mathcal{C} \setminus S$ is strictly harmed: $\mu'(j_0) \prec_{j_0} \mu(j_0)$.

Let $\mu(j_0) = h_l$ (the object j_0 receives under truthful TTC, which is l's endowment). Since j_0 reports truthfully in the misreport run and always picks their top available object, the fact that j_0 gets $\mu'(j_0) \prec_{j_0} h_l$ at round r^* means h_l is *not available* at the start of round r^* in the misreport run. So h_l was taken in some round $r' < r^*$ by some agent $m \neq j_0$.

We now derive a contradiction by tracing who took h_l:

- $m \notin S^+$: by choice of r^*, no S^+-member exits before round r^*. So $m \notin S^+$.
- $m \notin S^0$: if $m \in S^0$, then $\mu'(m) = \mu(m)$. In round r', m exits and gets h_l, so $\mu'(m) = h_l$. But then $\mu(m) = h_l = \mu(j_0)$, implying $m = j_0$ (since μ is a bijection). This contradicts $m \in S^0 \subseteq S$ and $j_0 \notin S$.
- Therefore $m \notin S$, and m reports truthfully. In the truthful run, m's exit cycle forms at round r_m (the round m exits under truth-telling). Under truthful TTC, m gets $\mu(m) \neq h_l$ (since $\mu^{-1}(h_l) = j_0 \neq m$ and μ is a bijection). So m is *not* in the cycle containing l in the truthful run: under truthful TTC, m does not point to l at the round where h_l changes hands.

So in the misreport run, the truthful agent $m \notin S$ ends up in a different cycle than in the truthful run, specifically one that includes l and yields h_l to m. Since m

always reports truthfully, m's pointing changes only if the available set at m's exit round changes. This change in the available set must itself be caused by some S-member's misreport altering an earlier cycle. Tracing this chain: let $\mathcal{C}'$ be the TTC cycle in round r' that contains m (and l) in the misreport run. By the same case analysis applied to $\mathcal{C}'$:

- If $\mathcal{C}' \subseteq S$: a sub-coalition of S is trading endowments in $\mathcal{C}'$. Since $m \notin S$ is in $\mathcal{C}'$, this is impossible.
- If $\mathcal{C}' \not\subseteq S$: $\mathcal{C}'$ contains some S-member. If all agents in $\mathcal{C}'$ weakly benefit (sub-case 2a for $\mathcal{C}'$), then $\mathcal{C}'$ is a blocking coalition—contradiction. If some agent in $\mathcal{C}' \setminus S$ is strictly harmed, the same tracing argument applies recursively to a strictly earlier round.

Since the rounds are finite and strictly decreasing at each recursive step, this process must terminate. The terminal case is either sub-case 2a (blocking coalition, contradiction with core) or Case 1 (coalition entirely within S, contradiction with core). Either way, we reach a contradiction.

Since all cases yield contradictions, no coalition S can achieve a weak Pareto improvement via coordinated misreporting. TTC is weakly group strategy-proof. ∎

Remark (Why Weak GSP Is Harder Than Strong GSP).

The additional difficulty in the weak GSP proof relative to strong GSP comes entirely from S^0-members. In strong GSP, every coalition member is assumed strictly better off, so individual strategy-proofness can be applied to each member to show they can only receive objects from within S (Claim 1 of the strong GSP proof). This argument fails for S^0-members: they are indifferent between their truthful and misreport outcomes, so ISP provides no constraint on where their object originates.

S^0-members may misreport not to benefit themselves, but to act as "facilitators"—adjusting the cycle structure in early rounds to create a path that allows S^+-members to enter better cycles. Sub-case 2b captures exactly this scenario and requires the recursive tracing argument above: each step of the chain that S^0-members create must eventually close into a cycle, and that cycle is forced to be a blocking coalition for μ under true preferences—which is impossible since μ is the unique strict core.

6.2 The Assignment Problem: Allocation without Endowments

The housing market model rests critically on the assumption that each agent holds an initial endowment: this is what gives IR its bite, and what makes the core a meaningful solution concept. Once we drop this assumption—handing all objects to a central planner and asking how to distribute them from scratch—both IR and the core lose their reference point. We enter the *assignment problem*, where no agent has a prior claim to any object and the relevant normative goals reduce to *Pareto efficiency* and *strategy-proofness*.

6.2.1 Serial Dictatorship (SD)

The serial dictatorship (SD) mechanism addresses this setting directly. Agents are given a predetermined priority ordering and pick their favorite available object in turn. Compared to TTC, SD trades the structural richness of cycle-based trading for sheer simplicity: there are no endowments to track, no cycles to identify, and no core to verify—just a queue and a sequence of greedy choices.

Serial Dictatorship

Fix an ordering $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(n))$ of the agents. The mechanism proceeds as follows:

1. Agent $\sigma(1)$ (the "dictator") picks their most preferred house from H.
2. Agent $\sigma(2)$ picks their most preferred house from the remaining houses.
3. Continue until all agents are assigned.

Example (Serial Dictatorship on the Office Problem).

We run SD on the office problem with the natural ordering $\sigma = (P_1, P_2, \ldots, P_9)$. At each step, the active agent scans their preference list top-to-bottom and picks the highest-ranked office that is still available. Taken offices are shown in gray.

Step	Available offices	Agent picks	Gets
1	$\{1,2,3,4,5,6,7,8,9\}$	P_1: top choice is 7 ✓	7
2	$\{1,2,3,4,5,6,8,9\}$	P_2: top choice is 4 ✓	4
3	$\{1,2,3,5,6,8,9\}$	P_3: top choice is 6 ✓	6
4	$\{1,2,3,5,8,9\}$	P_4: 7 taken; next available: 1 ✓	1
5	$\{2,3,5,8,9\}$	P_5: 7, 1 taken; next available: 8 ✓	8
6	$\{2,3,5,9\}$	P_6: 7, 1 taken; next available: 2 ✓	2
7	$\{3,5,9\}$	P_7: top choice is 9 ✓	9
8	$\{3,5\}$	P_8: 6, 2, 4 taken; next available: 3 ✓	3
9	$\{5\}$	P_9: 2 taken; next available: 5 ✓	5

The SD allocation under ordering $\sigma = (P_1, \ldots, P_9)$ is:

Agent	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Office	7	4	6	1	8	2	9	3	5

Compare this with the TTC allocation on the same problem:

Agent	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Office	6	4	8	7	5	1	9	3	2

Several agents differ: P_1 gets office 7 under SD (their top choice) versus 6 under TTC;

P_4 gets 1 under SD versus 7 under TTC. This illustrates a key tradeoff: the agent who goes first under SD always receives their top choice, but later agents may fare worse than under TTC, where the cycle structure can give non-first agents access to offices they could not reach by waiting their turn.

6.2.2 Properties of SD

Theorem 6.9: Serial Dictatorship is Pareto Efficient

The allocation produced by SD is Pareto efficient: there is no other allocation that makes every agent weakly better off and some agent strictly better off.

Proof for Theorem

The argument is essentially one of optimality at the margin. Consider the first agent i who would need to be reassigned to achieve the supposed Pareto improvement. Agent i already holds their top choice among everything not taken by higher-priority agents—there is simply nothing better left for them. Any Pareto improvement for i is therefore impossible, and the supposed dominating allocation cannot exist.

Suppose for contradiction that some allocation μ' Pareto dominates the SD allocation μ. Let $i = \sigma(k)$ be the *first* agent in the priority ordering such that $\mu'(i) \neq \mu(i)$. Since μ' Pareto dominates μ, we have $\mu'(i) \succ_i \mu(i)$.

But $\mu(i)$ is i's most preferred object among all objects available when it was i's turn—that is, among all objects not already taken by $\sigma(1), \ldots, \sigma(k-1)$. Since i is the first agent to differ between μ and μ', every agent $\sigma(1), \ldots, \sigma(k-1)$ receives the same object under both μ and μ'. Therefore the set of objects available to i at step k is identical under both allocations, and $\mu(i)$ is i's top choice from that set. The existence of $\mu'(i) \succ_i \mu(i)$ with $\mu'(i)$ in the same available set is a contradiction.

*Notice that this argument does **not** require every agent to get their globally top-ranked object. It only requires that each agent gets their top choice **given the residual supply**, which is precisely what the greedy sequential structure of SD guarantees.* ■

Theorem 6.10: Serial Dictatorship is Strategy-Proof

Under SD, truthful reporting of preferences is a dominant strategy for every agent. No agent can obtain a strictly better object by misreporting their preferences, regardless of what other agents report.

Proof for Theorem

In SD, your report only affects what you pick*—it has no effect whatsoever on what is available to you, since the available set is determined entirely by agents ahead of you in the queue. You always face the same menu A_k, and truth-telling simply ensures you pick your genuine favorite from it. Lying can only cause you to pick something you like less.*

Fix any agent $i = \sigma(k)$ and any reports by all other agents. Suppose i misreports, yielding object o' under the misreport, compared to $o^* = \mu(i)$ under truthful reporting. We show $o' \not\succ_i o^*$.

The key observation is that, the set of objects available to i at step k is determined entirely by the choices of $\sigma(1), \ldots, \sigma(k-1)$, each of whom picks their own top available object according to their own report. Since $i = \sigma(k)$ has no influence over any earlier agent's report or choice, the available set A_k at step k is **identical** whether i reports truthfully or not.

Under truthful reporting, i picks $o^* = \mu(i)$, their most preferred object in A_k (the available set at step k). Under misreport, i picks according to a false preference list, and receives some $o' \in A_k$. Since o^* is i's true top choice from A_k and $o' \in A_k$, we have $o^* \succsim_i o'$. In particular, $o' \not\succ_i o^*$. Since the reports of other agents and the choice of $i = \sigma(k)$ were arbitrary, truth-telling is a dominant strategy. ∎

6.2.3 Equivalence of Random SD and TTC with Random Endowments

The separation between the housing market (with endowments, TTC) and the assignment problem (without endowments, SD) may seem sharp. Yet there is a striking bridge between the two frameworks: once we introduce randomness into either model in the natural way, the two mechanisms become outcome-equivalent. This result, due to Abdulkadiroğlu and Sönmez (1998), reveals that the endowment structure and the priority structure are, at a deeper level, two sides of the same coin.

The key observation is that both frameworks have one remaining degree of freedom that the model leaves unspecified:

- In the *assignment problem*, objects have no owner, so the mechanism designer must choose a **priority ordering** over agents. If this choice seems arbitrary, a natural response is to randomize: draw a uniformly random ordering.
- In the *housing market*, agents must have initial endowments, but if no agent has a natural prior claim to any particular object, a natural response is again to randomize: draw a uniformly random endowment assignment, then run TTC.

Randomizing the one free parameter in each model yields two randomized mechanisms. The theorem below says they are the same mechanism.

Definition 6.11: Randomized Mechanisms

- **Random Serial Dictatorship (RSD):** Draw a uniformly random ordering σ of agents (each of the $n!$ orderings equally likely), then run SD under σ.
- **Core from Random Endowments (CFRE):** Draw a uniformly random assignment of objects to agents as initial endowments (each of the $n!$ bijections equally likely), then run TTC to find the core allocation.

Theorem 6.12: RSD = CFRE

RSD and CFRE induce the *same* probability distribution over allocations: for every deterministic allocation μ,

$$\Pr_\sigma\big[\mu_\sigma^{\mathrm{SD}} = \mu\big] \;=\; \Pr_e\big[\mu_e^{\mathrm{TTC}} = \mu\big],$$

where σ is a uniformly random priority ordering and e is a uniformly random endowment assignment.

Proof for Theorem

Lemma 6.13: Within-Cycle Commutativity

Fix any endowment e and let $\mu = \mu_e^{\mathrm{TTC}}$, with TTC producing cycles $C_1, C_2, \ldots, C_m$ in successive rounds. Let $\Sigma(\mu, e)$ denote the set of all priority orderings that:

1. place all agents in C_k before all agents in C_{k+1} (for each k), and
2. within each C_k, use any permutation of its members.

Then SD under every $\sigma \in \Sigma(\mu, e)$ produces μ.

Proof for Lemma

We establish one fact about the TTC cycle structure.

Fact 6.14: Within-Cycle Self-Preference

For any cycle C_k and any two distinct members $a, b \in C_k$: $\mu(a) \succ_a \mu(b)$.

Take any $\sigma \in \Sigma(\mu, e)$ and any agent $a \in C_k$. When a's turn comes in SD, the available objects are a subset of those available in TTC round k, for the following reason: the objects removed before round k in TTC are exactly $\bigcup_{l<k}\{\mu(x) : x \in C_l\}$, and by definition of $\Sigma(\mu, e)$, all agents in $C_1, \ldots, C_{k-1}$ pick before a. By the Fact above and induction, each such agent picks their own μ-assignment, so the objects removed before a's turn in SD are exactly $\bigcup_{l<k}\{\mu(x) : x \in C_l\}$ plus the μ-assignments of those C_k-members who pick before a.

We verify that $\mu(a)$ is a's top available object when a picks:

- **Objects from earlier cycles are unavailable.** All $\mu(i)$ for $i \in C_l$, $l < k$ have been claimed. In TTC, $\mu(a)$ is a's top choice among all objects available in round k—this already accounts for the absence of earlier objects. So no unavailable object is preferred over $\mu(a)$.
- $\mu(a)$ **has not been taken by earlier** C_k **members.** By the Fact, every other $b \in C_k$ prefers $\mu(b)$ over $\mu(a)$, so no earlier C_k-member claims $\mu(a)$.
- **Objects from later cycles are weakly worse.** All objects $\mu(x)$ for $x \in C_l$, $l > k$ were available in TTC round k, and a chose $\mu(a)$ over all of them.

Therefore a claims $\mu(a)$ in SD. Since a was arbitrary, SD under σ produces μ. ∎

It remains to show that the randomization in CFRE and RSD induce the same distribution. We do this by mapping both the valid orderings σ and the valid endowments e to a canonical block partition uniquely defined by μ and the agents' preferences.

Fix an allocation μ. Define an ordered partition $\mathcal{B} = (B_1, B_2, \ldots, B_m)$ of the agents recursively as follows: Let B_1 be the minimal nonempty set of agents whose top choices among all objects are exactly $\mu(B_1)$. Let B_2 be the minimal nonempty set of agents whose top choices among the remaining objects $N \setminus \mu(B_1)$ are exactly $\mu(B_2)$, and so on. Notice that $\mathcal{B}$ is uniquely determined independent of σ or e.

Counting Orderings for RSD:
An ordering σ yields μ under SD if and only if it respects this dependency structure. Specifically, agents in B_k must appear after all agents in $B_1, \ldots, B_{k-1}$ to ensure their preferred objects in those blocks are already taken. Because B_k is defined as the minimal closed set under top available choices, the agents within B_k can be processed in *any* internal order without disrupting the assignment μ. Thus, the total number of orderings yielding μ is exactly the number of ways to permute agents within each block:

$$\left|\{\sigma : \mu_\sigma^{\mathrm{SD}} = \mu\}\right| = \prod_{k=1}^{m} |B_k|!$$

Counting Endowments for CFRE:
An endowment e yields μ under TTC if and only if for every block k, e endows the objects $\mu(B_k)$ strictly to the agents in B_k (i.e., $e(B_k) = \mu(B_k)$). To see why, consider the TTC pointer function $p(i) = e^{-1}(\mu(i))$. If $e(B_k) = \mu(B_k)$, then p forms a bijection from B_k to itself. Consequently, p perfectly decomposes into a set of disjoint cycles covering all of B_k. Since their top available choices are in $\mu(B_k)$, every agent $i \in B_k$ participates in a cycle and receives exactly $\mu(i)$.

The number of such valid endowments is simply the number of possible bijections from B_k to $\mu(B_k)$ for each block k. Therefore:

$$\left|\{e : \mu_e^{\mathrm{TTC}} = \mu\}\right| = \prod_{k=1}^{m} |B_k|!$$

Since there are exactly $n!$ total orderings and $n!$ total endowments, and the sizes of the pre-images yielding μ are strictly equal, we conclude:

$$\Pr_\sigma\left[\mu_\sigma^{\mathrm{SD}} = \mu\right] = \frac{\prod_{k=1}^{m} |B_k|!}{n!} = \Pr_e\left[\mu_e^{\mathrm{TTC}} = \mu\right]$$

This holds for any deterministic allocation μ, completing the proof. ∎

Remark (Proof Intuition).

The bijection has a clean interpretation. In SD, the priority ordering determines who gets served first. In TTC, the endowment determines who holds what. The bijection $\sigma \mapsto e(\sigma)$ encodes the SD priority order *into* the endowment structure by a cyclic shift: agent σ_k is endowed with the object that σ_{k-1} ends up with under SD. This creates a pointing chain in TTC that exactly mirrors the SD queue—agent σ_k points to σ_{k+1} because σ_{k+1} holds what σ_k ultimately wants. The TTC cycles that form then execute the same trades as the SD steps, in the same order.

At a higher level, both mechanisms are doing the same thing: translating an underlying permutation (of agents, or of objects) into an allocation by a greedy sequential procedure. Randomizing the permutation uniformly in either case produces the same lottery, because the two "greedy procedures" are isomorphic via the cyclic shift bijection.

The equivalence has several consequences worth noting.

- **Mechanism design flexibility.** A planner who knows only agents' preferences—but has no natural endowment structure or priority structure—can implement the same outcome-distribution using whichever model is more convenient for the context.
- **Properties transfer.** Since RSD inherits strategy-proofness and Pareto efficiency from SD, and CFRE inherits the core property and IR from TTC, the equivalence implies that both randomized mechanisms simultaneously satisfy all four properties in expectation.
- **The endowment is not deep.** The result suggests that the distinction between "having endowments" and "not having endowments" is less fundamental than it appears—at least

once both are randomized. What matters is the underlying combinatorial structure of sequential greedy assignment, not the narrative framing of endowments versus priorities.

6.3 Two-Sided Matching

We now transition from one-sided allocation problems (housing markets) to *two-sided matching* problems, where agents on both sides of the market have preferences over agents on the other side. The canonical example is the *college admissions* problem introduced by Gale and Shapley (1962).

6.3.1 Two-Sided Matching and Pairwise Stability

Definition 6.15: Two-Sided Matching Market

A **two-sided matching market** (or *marriage model*) consists of:

- Two disjoint, finite sets of agents: $\mathcal{S} = \{s_1, \ldots, s_n\}$ (students) and $\mathcal{C} = \{c_1, \ldots, c_m\}$ (colleges).
- Each student $s \in \mathcal{S}$ has a strict preference ordering $\succ_s$ over $\mathcal{C}$.
- Each college $c \in \mathcal{C}$ has a strict preference ordering $\succ_c$ over $\mathcal{S}$.
- Each college has a **quota** $q_c \geq 1$ specifying the number of seats available. In the basic model, $q_c = 1$ for all c, and $m = n$.

Definition 6.16: Matching

A **matching** μ is a bijection $\mu : \mathcal{C} \to \mathcal{S}$ (when $|\mathcal{C}| = |\mathcal{S}|$ and all quotas are 1). That is, $\mu(c)$ denotes the student assigned to college c, and $\mu^{-1}(s)$ denotes the college assigned to student s.

In the housing market (Shapley–Scarf), only one side—the agents—had preferences; houses were passive objects. Here, *both* sides have preferences, which fundamentally changes the problem. The notion of "core" from the housing market generalizes to the concept of *stability*, but the structure of stable allocations is much richer.

The central solution concept in two-sided matching is *pairwise stability*. Unlike the core (which considers arbitrary coalitions), pairwise stability focuses on deviations by pairs (c, s).

Definition 6.17: Blocking Pair and Pairwise Stability

A pair $(c, s) \in \mathcal{C} \times \mathcal{S}$ is a **blocking pair** for matching μ if:

1. College c prefers student s to its current match: $s \succ_c \mu(c)$;
2. Student s prefers college c to their current match: $c \succ_s \mu^{-1}(s)$.

A matching μ is **pairwise stable** if there exists no blocking pair.

Remark.

Pairwise stability captures a minimal notion of "**no justified envy**": there is no student-college pair who would both prefer to be matched with each other over their current assignments. If such a pair existed, the college would have an incentive to admit the student, and the student would have an incentive to accept—destabilizing the matching.

6.3.2 Deferred Acceptance (DA)

Gale and Shapley (1962) introduced the *Deferred Acceptance* (DA) algorithm, which always produces a pairwise stable matching.

Student-Proposing Deferred Acceptance (DA)

1. **Propose:** Each unmatched student proposes to the highest-ranked college on their list to which they have not yet proposed.
2. **Hold/Reject:** Each college that receives proposals *tentatively* holds the most preferred applicant (according to its own ranking) among the new proposals and its currently held student (if any), and rejects all others.
3. **Repeat:** Rejected students propose to their next choice. Continue until no rejections occur (every student is either tentatively held or has been rejected by all colleges).
4. **Finalize:** All tentative matches become permanent.

Example (Student-Proposing DA on the Matching Problem).

Consider 10 students $S = \{s_1, \ldots, s_{10}\}$ and 10 colleges $C = \{c_1, \ldots, c_{10}\}$, each with quota $q = 1$.

College preferences over students (Final matches highlighted):

Rank	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
1	s_1	s_1	s_{10}	s_6	s_7	**s_4**	s_1	s_6	s_8	s_6
2	s_2	s_6	s_2	s_1	s_2	s_8	s_4	s_4	**s_{10}**	s_5
3	**s_3**	s_9	s_4	s_5	s_8	s_2	s_3	s_2	s_6	**s_1**
4	s_4	s_2	**s_6**	**s_2**	s_4	s_6	s_8	s_{10}	s_9	s_4
5	s_5	s_5	s_8	s_7	s_{10}	s_{10}	**s_5**	s_8	s_5	s_{10}
6	s_6	**s_8**	s_3	s_3	s_5	s_3	s_2	s_3	s_7	s_8
7	s_7	s_3	s_5	s_8	**s_9**	s_7	s_9	s_1	s_4	s_9
8	s_8	s_4	s_7	s_4	s_6	s_1	s_6	s_9	s_3	s_2
9	s_9	s_7	s_9	s_{10}	s_3	s_5	s_7	**s_7**	s_1	s_3
10	s_{10}	s_{10}	s_1	s_9	s_1	s_9	s_{10}	s_5	s_2	s_7

Student preferences over colleges (Final matches highlighted):

Rank	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
1	c_5	**c_4**	c_5	**c_6**	c_3	**c_3**	**c_8**	c_1	c_1	c_2
2	**c_{10}**	c_9	c_6	c_2	**c_7**	c_4	c_6	c_4	**c_5**	c_{10}
3	c_6	c_{10}	**c_1**	c_5	c_9	c_5	c_4	**c_2**	c_2	**c_9**
4	c_4	c_6	c_7	c_8	c_{10}	c_1	c_2	c_6	c_{10}	c_4
5	c_7	c_3	c_4	c_1	c_1	c_9	c_9	c_3	c_3	c_1
6	c_2	c_7	c_8	c_7	c_5	c_7	c_7	c_7	c_8	c_7
7	c_8	c_1	c_9	c_3	c_4	c_{10}	c_5	c_5	c_6	c_5
8	c_9	c_5	c_{10}	c_9	c_2	c_8	c_{10}	c_{10}	c_7	c_6
9	c_1	c_2	c_2	c_{10}	c_6	c_2	c_1	c_8	c_4	c_3
10	c_3	c_8	c_3	c_4	c_8	c_6	c_3	c_9	c_9	c_8

We execute the student-proposing DA algorithm. We track the process using a chalkboard-style simulation: the colleges are listed across the top, and proposing students are written below them. In each round, if a college receives multiple proposals (or a new proposal challenges a held student), the college's less preferred student is simply crossed out (~~struck through~~).

Round 1.
All 10 students propose to their first-choice colleges. Colleges c_1, c_3, and c_5 receive multiple proposals and must reject the less preferred students.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_8	s_{10}	s_6	s_2	s_3	s_4		s_7		
~~s_9~~		~~s_5~~		~~s_1~~					

Rejected: s_9 (next c_5), s_5 (next c_7), s_1 (next c_{10}).

Round 2.
The three rejected students propose to their next choices. s_9 challenges s_3 at c_5. College c_5 prefers s_9 (#7) over s_3 (#9).

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_8	s_{10}	s_6	s_2	s_9	s_4	s_5	s_7		s_1
				~~s_3~~					

Rejected: s_3 (next c_6).

Round 3.
s_3 proposes to c_6. College c_6 evaluates the held s_4 and the new s_3. s_4 is c_6's top pick and is retained.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_8	s_{10}	s_6	s_2	s_9	s_4	s_5	s_7		s_1
					~~s_3~~				

Rejected: s_3 (next c_1).

Round 4.

s_3 proposes to c_1. College c_1 prefers s_3 (#3) over the currently held s_8 (#8).

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_{10}	s_6	s_2	s_9	s_4	s_5	s_7		s_1
~~s_8~~									

Rejected: s_8 (next c_4).

Round 5.

s_8 proposes to c_4. College c_4 prefers the held s_2 (#4) over s_8 (#7).

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_{10}	s_6	s_2	s_9	s_4	s_5	s_7		s_1
			~~s_8~~						

Rejected: s_8 (next c_2).

Round 6.

s_8 proposes to c_2. College c_2 prefers s_8 (#6) over the held s_{10} (#10).

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_8	s_6	s_2	s_9	s_4	s_5	s_7		s_1
	~~s_{10}~~								

Rejected: s_{10} (next c_{10}).

Round 7.

s_{10} proposes to c_{10}. College c_{10} prefers the held s_1 (#3) over s_{10} (#5).

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_8	s_6	s_2	s_9	s_4	s_5	s_7		s_1
									~~s_{10}~~

Rejected: s_{10} (next c_9).

Round 8.

s_{10} proposes to c_9. College c_9 is empty and accepts immediately.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_8	s_6	s_2	s_9	s_4	s_5	s_7	s_{10}	s_1

No further rejections. The algorithm terminates.

The **student-optimal stable matching** μ_S is directly read from the final chalkboard state:

College	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Student	s_3	s_8	s_6	s_2	s_9	s_4	s_5	s_7	s_{10}	s_1

6.3.3 Properties of DA

Theorem 6.18: DA is Pairwise Stable

The matching produced by the student-proposing DA algorithm is pairwise stable.

Proof for Theorem

The argument proceeds as follows. Suppose, for contradiction, that (c, s) is a blocking pair under the DA outcome μ. Then:

1. Student s prefers c to $\mu^{-1}(s)$, meaning $c \succ_s \mu^{-1}(s)$. Since DA has students propose in order of preference, s must have proposed to c at some earlier round before being eventually matched with $\mu^{-1}(s)$.

2. But s was rejected by c at that round, which means c was holding (tentatively matched with) a student it prefers to s. Since colleges only ever improve their tentative match over the course of the algorithm, c's final match $\mu(c)$ is at least as preferred as the student it held when it rejected s. Hence $\mu(c) \succ_c s$.

This contradicts the requirement that c prefers s to $\mu(c)$. Therefore no blocking pair can exist. ∎

Theorem 6.19: Strategy-Proofness for the Proposing Side

In the student-proposing DA, truthful reporting of preferences is a dominant strategy for every **student**. That is, no student can obtain a strictly better outcome by misreporting their preferences, regardless of what other students or colleges report.

Proof for Theorem

We present the proof for the general many-to-one matching market where each college c has a quota $q_c \geq 1$.

We proceed by contradiction. Suppose there exists a true preference profile P, but some student s^* submits a fake preference list P'_{s^*}. Let μ be the truthful DA outcome, and let μ' be the DA outcome under the misreported profile $P' = (P'_{s^*}, P_{-s^*})$.

Suppose the misreport is strictly profitable for s^*, meaning $\mu'(s^*) \succ_{s^*} \mu(s^*)$.

Step 1: Identify the "Improving" Coalition.
Let S' be the set of all students who are strictly better off under μ' than under μ according to their true preferences:

$$S' = \{s \in \mathcal{S} \mid \mu'(s) \succ_s \mu(s)\}$$

By our assumption, $s^* \in S'$. Let $C' = \mu'(S')$ be the set of colleges these students are matched to in μ'.

Step 2: The Rejection Implication.
For every student $s \in S'$, because $\mu'(s) \succ_s \mu(s)$, s must have proposed to $\mu'(s)$ and been rejected during the *truthful* DA. Consequently, every college $c \in C'$ must have rejected at least one student from S' in the truthful DA.

Under the rules of DA with quotas, a college only rejects an acceptable student if

its quota is completely full. This implies that in the truthful outcome μ, every college $c \in C'$ exactly filled its capacity ($|\mu(c)| = q_c$), and it strictly prefers **every single student** currently in $\mu(c)$ to the S' student it rejected.

Step 3: All Displaced Students must belong to S'.
Take any college $c \in C'$. Since $c \in C'$, there is some student $s \in S'$ such that $s \in \mu'(c)$. Let $k \in \mu(c)$ be **any** student assigned to c in the truthful DA. We claim k must also belong to S'.

Suppose for contradiction that $k \notin S'$. This means k does not strictly prefer μ' over μ. Since c must be filled with different students or drop some students to accommodate s in μ', if k is displaced from c, k strictly prefers $\mu(k) = c$ over $\mu'(k)$. Since $k \notin S'$ but $s^* \in S'$, $k \neq s^*$. Thus, k reported truthfully.

Now consider the matching μ' under the reported profile P'. College c admits $s \in S'$ in μ'. But from Step 2, c strictly prefers k over s ($k \succ_c s$). At the same time, k strictly prefers c over $\mu'(k)$. Because k's reported preference is their true preference, (c, k) forms a **blocking pair** for μ' under P'! This contradicts the fact that μ' is the stable DA outcome under P'.

Hence, our claim holds: **every** student $k \in \mu(c)$ sitting in college c during the truthful DA must belong to S'.

Step 4: The Counting Contradiction.
Let us count the seats in C'. The total capacity of all colleges in C' is $\sum_{c \in C'} q_c$.

By Step 2, every college in C' is completely full under μ. By Step 3, every single student occupying these seats in μ belongs to S'. Therefore, S' contains **at least** $\sum_{c \in C'} q_c$ students.

Now look at μ'. By the definition of C', every student in S' is assigned to a college in C'. Since the maximum number of students C' can hold is $\sum_{c \in C'} q_c$, S' can contain **at most** $\sum_{c \in C'} q_c$ students.

Thus, $|S'| = \sum_{c \in C'} q_c$. The students assigned to C' in the truthful matching μ are *exactly* the students in S', occupying all available seats.

In μ', this exact same group of students S' is completely reallocated among the exact same seats in C'. But by the definition of S', **every single student in S' strictly prefers their new seat in μ' to their old seat in μ.**

This implies μ' **strictly Pareto dominates** μ for this subset of students, without affecting anyone else. However, it is a fundamental property of the student-proposing DA that its outcome (μ) is the **student-optimal stable matching** (which is Pareto efficient for the students). A strict Pareto improvement via a pure internal reallocation is mathematically impossible.

This contradiction proves that no such profitable misreport P'_{s^*} can exist. ∎

Remark.

The intuition is clean: under DA, a student who misreports can only cause themselves to be rejected by colleges they would have been admitted to, or cause themselves to propose to less preferred colleges earlier. Misreporting can never lead to admission at a college that would have rejected the student under truthful reporting.

Theorem 6.20: Impossibility of Two-Sided Strategy-Proofness

No stable matching mechanism is strategy-proof for **both** sides simultaneously. In particular, under the student-proposing DA, colleges may have incentives to misreport their preferences.

Remark (How Can the Accepting Side Manipulate?).

The accepting side (colleges in student-proposing DA) can potentially manipulate by *truncating* their preference lists or *reordering* them. For instance, a college might strategically rank a less-preferred student higher to trigger a chain of rejections that ultimately results in a more favorable match. This impossibility result (Roth, 1982) highlights a fundamental tension: **any stable matching mechanism must sacrifice strategy-proofness on at least one side**.

6.3.4 Many-to-One Matching (Quotas)

So far, our two-sided matching model assumed $q_c = 1$ for all colleges, mathematically mirroring a strict one-to-one "marriage" market. In reality, the classic Gale-Shapley model was built for the *college admissions* problem, where institutions admit entire incoming classes. Generalizing from one-to-one to many-to-one matching introduces capacity constraints, but the core theoretical machinery translates beautifully.

Definition 6.21: Many-to-One Matching Market

Each college $c \in \mathcal{C}$ now has a maximum capacity or **quota** $q_c \geq 1$. A **matching** μ is a mapping such that:

- For each student s, $\mu(s) \in \mathcal{C} \cup \{\emptyset\}$ (a student attends at most one college or remains unmatched).
- For each college c, $\mu(c) \subseteq \mathcal{S}$ with $|\mu(c)| \leq q_c$ (a college admits a *set* of students up to its quota).
- $\mu(s) = c \iff s \in \mu(c)$.

In a many-to-one matching market, a college c with quota q_c ultimately admits a *set* of students. While the model specifies the college's strict ranking over individual students (denoted by $\succ_c$), we must define how it ranks different incoming classes (subsets of students). To keep the model tractable and rule out complex complementarities (e.g., "I only want student A if I also get student B"), we impose a standard behavioral assumption on how colleges evaluate groups.

Assumption 6.22: Responsive Preferences

Let $\succ_c^*$ denote college c's preference relation over *sets* of students. We assume $\succ_c^*$ is **responsive** to the college's preferences over individual students $\succ_c$. Specifically, for any set of students S strictly below capacity ($|S| < q_c$), and any two students $s, s' \notin S$:

1. **Substitution:** Replacing a student with a strictly more preferred individual always makes the college strictly better off:
$$S \cup \{s\} \succ_c^* S \cup \{s'\} \iff s \succ_c s'$$
2. **Addition:** Adding an acceptable student (one who is preferred to leaving a seat empty, $s \succ_c \emptyset$) always makes the college strictly better off:
$$S \cup \{s\} \succ_c^* S \iff s \succ_c \emptyset$$

Remark (Why do we need this assumption?).

Without responsiveness, evaluating stability becomes a combinatorial nightmare. A college might reject a globally top-ranked student simply because they do not "fit" well with the rest of the admitted cohort.

By assuming responsive preferences, we guarantee that a college's evaluation of a group is strictly and additively tied to the independent quality of its individual members. *Crucially, this mathematically justifies why we only need to check for pairwise blocking pairs.* If a college ever wants to deviate and alter its admitted class, responsiveness ensures that this desire will manifest as a simple pairwise blocking condition with a single better student, relieving us from checking complex multi-student coordinated deviations.

Definition 6.23: Pairwise Stability with Quotas

A matching μ is pairwise stable if it is individually rational and contains no **blocking pair**. A pair (c, s) blocks μ if:

1. Student s prefers c to their current match: $c \succ_s \mu(s)$;
2. **Either** c has an empty seat ($|\mu(c)| < q_c$),
Or c is full ($|\mu(c)| = q_c$) but prefers s to at least one of its currently admitted students ($s \succ_c s'$ for some $s' \in \mu(c)$).

Student-Proposing DA (with Quotas)

The DA algorithm adapts to quotas with a single, straightforward modification:

1. **Propose:** Each unassigned student proposes to their most preferred acceptable college that has not yet rejected them.

2. **Hold/Reject:** Each college c considers its new applicants alongside the students it tentatively held from the previous round. It ranks this combined pool, **tentatively holds up to q_c of its most preferred students**, and rejects the rest.

3. **Repeat** until no student is rejected.

Example (DA with Quotas).

Consider 6 students $\{s_1, \dots, s_6\}$ and 2 colleges $\{c_1, c_2\}$. College c_1 has a quota $q_1 = 2$, and college c_2 has a quota $q_2 = 3$. Notice that total demand (6 students) exceeds total supply (5 seats), meaning at least one student will be left unmatched.

College preferences (Quotas in parentheses, Final matches highlighted):

Rank	c_1 $(q = 2)$	c_2 $(q = 3)$
1	$\boldsymbol{s_1}$	$\boldsymbol{s_6}$
2	$\boldsymbol{s_2}$	$\boldsymbol{s_5}$
3	s_3	$\boldsymbol{s_4}$
4	s_4	s_3
5	s_5	s_2
6	s_6	s_1

Student preferences (Final matches highlighted):

Rank	s_1	s_2	s_3	s_4	s_5	s_6
1	c_2	$\boldsymbol{c_1}$	c_1	c_1	$\boldsymbol{c_2}$	c_1
2	$\boldsymbol{c_1}$	c_2	c_2	$\boldsymbol{c_2}$	c_1	$\boldsymbol{c_2}$

We run the student-proposing DA algorithm. Remember that colleges rank their combined pool of applicants and simply draw a "cutoff line" at their quota.

Round 1.
Students propose to their top choices: $s_1, s_5 \to c_2$, while $s_2, s_3, s_4, s_6 \to c_1$.

c_1 $(q = 2)$	c_2 $(q = 3)$
s_2	s_1
s_3	s_5
~~s_4~~	
~~s_6~~	

Explanation: c_1 receives 4 applicants for 2 seats. Looking at c_1's preference list, $s_2 \succ s_3 \succ s_4 \succ s_6$. It holds the top two (s_2, s_3) and rejects the rest. c_2 only has 2 applicants for 3 seats, so it holds both.
Rejected: s_4 (next c_2), s_6 (next c_2).

Round 2.
The rejected students s_4 and s_6 propose to c_2. c_2's new pool is $\{s_1, s_5\}$ (held) $\cup$ $\{s_4, s_6\}$ (new) $= \{s_1, s_4, s_5, s_6\}$.

$c_1\,(q=2)$	$c_2\,(q=3)$
s_2	s_6
s_3	s_5
	s_4
	~~s_1~~

Explanation: c_2 ranks its pool: $s_6 \succ s_5 \succ s_4 \succ s_1$. It keeps the top 3. Notice that s_1, who was safely held in Round 1, is now bumped out by stronger applicants!
Rejected: s_1 (next c_1).

Round 3.
s_1 proposes to c_1. c_1's pool is now $\{s_2, s_3\}$ (held) $\cup$ $\{s_1\}$ (new).

$c_1\,(q=2)$	$c_2\,(q=3)$
s_1	s_6
s_2	s_5
~~s_3~~	s_4

Explanation: c_1 ranks its pool: $s_1 \succ s_2 \succ s_3$. It keeps the top 2. Another devastating bump occurs: s_3 is kicked out to make room for s_1.
Rejected: s_3 (next c_2).

Round 4.
s_3 proposes to c_2. c_2's pool is $\{s_4, s_5, s_6\}$ (held) $\cup$ $\{s_3\}$ (new).

$c_1\,(q=2)$	$c_2\,(q=3)$
s_1	s_6
s_2	s_5
	s_4
	~~s_3~~

Explanation: c_2 ranks its pool: $s_6 \succ s_5 \succ s_4 \succ s_3$. The cutoff is 3, so s_3 is rejected immediately.
Rejected: s_3 (no choices remaining).

Round 5.
Student s_3 has exhausted their preference list. No unassigned student has an acceptable college left to propose to. The algorithm terminates.

The final student-optimal stable matching μ_S with quotas is:

$$\mu_S(c_1) = \{s_1, s_2\}, \quad \mu_S(c_2) = \{s_4, s_5, s_6\}, \quad \mu_S(s_3) = \emptyset$$

6.3.5 Truncated Preferences (Unacceptable Matches)

In our previous examples, we implicitly assumed that every college was acceptable to every student, and every student was acceptable to every college. In reality, matching markets

are characterized by **outside options**. A student might prefer to stay home or enter the labor market rather than attend a poorly ranked college. Similarly, a college might prefer to leave a seat empty rather than admit a student who does not meet its minimum academic standards.

We model this by introducing **truncated preferences**. An agent ranks a subset of the opposite side and deems the rest *unacceptable*.

Definition 6.24: Unacceptable Matches, Matching with Truncations and Individual Rationality

We use the symbol $\emptyset$ to denote the outside option of remaining unmatched.

- A college c is **acceptable** to student s if $c \succ_s \emptyset$.
- A student s is **acceptable** to college c if $s \succ_c \emptyset$.

Any agent ranked below $\emptyset$ in a preference list is strictly worse than being unmatched.

Definition 6.25: Matching with Truncations

Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ (with quotas q_j) and $\mathcal{S} = \{s_1, \ldots, s_n\}$. Each student s has a strict preference $\succ_s$ over $\mathcal{C}$, and each college c has a strict preference $\succ_c$ over $\mathcal{S}$. Agents may rank only a subset of the other side as acceptable, leaving the ranking below $\emptyset$ as unacceptable. A **matching** is a function $\mu : \mathcal{S} \to \mathcal{C} \cup \{\emptyset\}$ (with the constraint that $|\{s : \mu(s) = c\}| \leq q_c$ for each c). Here $\mu(s) = \emptyset$ means student s is unmatched.

Definition 6.26: Individual Rationality

A matching μ is **Individually Rational (IR)** if no agent is matched with an unacceptable partner: $\mu(s) \succsim_s \emptyset$ and $\mu(c) \succsim_c \emptyset$ for all $s \in \mathcal{S}, c \in \mathcal{C}$.

Definition 6.27: Stability with Truncations

A matching μ is **stable** if:

(i) **Individual rationality:** Every matched student prefers their match to being unmatched: if $\mu(s) \in \mathcal{C}$, then $\mu(s) \succ_s \emptyset$. Similarly, every college finds each of its matched students acceptable.

(ii) **No blocking pair:** There is no pair (s, c) with $\mu(s) \neq c$ such that $c \succ_s \mu(s)$ and either c has an unfilled seat or $s \succ_c s'$ for some $s' \in \mu(c)$.

The DA algorithm handles truncated preferences effortlessly by embedding the IR constraint directly into its rules:

1. Students never propose to unacceptable colleges (they drop out of the market once they reach $\emptyset$ on their lists).

2. Colleges immediately reject any proposing student who is unacceptable to them, even if they have empty seats.

This slight modification is not just algorithmic; it provides a constructive proof for a fundamental existence theorem in matching theory.

Theorem 6.28: Existence of Stable Matchings with Truncations

For any two-sided matching market with strict, truncated preferences, there always exists at least one matching that is both pairwise stable and individually rational.

Proof for Theorem

The proof is constructive. We show that the Deferred Acceptance (DA) algorithm, modified with the two IR rules above, always terminates and outputs a matching that satisfies both properties.

Part 1: Individual Rationality (IR).
By the modified rules of the DA algorithm, no student ever proposes to an unacceptable college. Furthermore, no college ever tentatively holds (or permanently admits) an unacceptable student. Therefore, in the final matching μ, every student either receives an acceptable college or remains unmatched ($\mu(s) \succsim_s \emptyset$), and every college only admits acceptable students. Thus, μ is individually rational.

Part 2: Pairwise Stability.
Suppose, for contradiction, that there exists a blocking pair (c, s) in the final matching μ. By definition of a blocking pair under quotas and truncations:

1. Student s strictly prefers c over their final match: $c \succ_s \mu(s) \succsim_s \emptyset$.
2. College c strictly prefers s over $\emptyset$ (meaning s is acceptable to c), AND either c has an empty seat ($|\mu(c)| < q_c$) or c strictly prefers s over some admitted student $s' \in \mu(c)$.

Since $c \succ_s \mu(s)$, student s must have proposed to college c at some round prior to proposing to $\mu(s)$ (or before dropping out, if $\mu(s) = \emptyset$).

Since s is not matched with c in the end, college c must have rejected s at some round. Under the DA rules, a college only rejects an *acceptable* student if it has already filled its entire quota q_c with students it strictly prefers to the rejected student.

Moreover, as the DA algorithm progresses, a college's pool of tentatively held students can only improve (a college only drops a held student if a strictly better one proposes). Therefore, in the final matching μ, college c must be exactly at full capacity ($|\mu(c)| = q_c$), and it must strictly prefer every student in $\mu(c)$ to s.

This directly contradicts the second condition of the blocking pair (that c has an empty seat or prefers s to some admitted student s').

Hence, no blocking pair can exist. The modified DA algorithm always terminates yielding an individually rational and pairwise stable matching, proving existence. ■

Example (DA with Truncated Preferences).

Consider our 10-student, 10-college market again, but now with severe truncations. The symbol $\emptyset$ marks the end of the acceptable options. (For columns that do not explicitly show $\emptyset$, we assume the list naturally ends there or the unlisted options are unacceptable).

College preferences (Final matches highlighted):

Rank	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
1	s_1	s_1	s_{10}	s_6	s_7	**s_4**	s_1	s_6	s_8	s_6
2	s_2	s_6	s_2	s_1	s_2	s_8	s_4	s_4	**s_{10}**	**s_5**
3	**s_3**	**s_9**	s_4	s_5	s_8	s_2	s_3	s_2	s_6	s_1
4	s_4	s_2	**s_6**	**s_2**	s_4	s_6	**s_8**	s_{10}	s_9	s_4
5	s_5	$\emptyset$	s_8	s_7	$\emptyset$	s_{10}	s_5	s_8	s_5	s_{10}
6	s_6		s_3	s_3		s_3	s_2	s_3	s_7	s_8
7	$\emptyset$		s_5	s_8		s_7	s_9	s_1	s_4	s_9
8			s_7	s_4		s_1	s_6	$\emptyset$	s_3	s_2
9			s_9	s_{10}		s_5	s_7		s_1	s_3
10			s_1	s_9		s_9	s_{10}		s_2	s_7
11			$\emptyset$	$\emptyset$						

Student preferences (Final matches highlighted):

Rank	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
1	c_5	**c_4**	c_5	**c_6**	c_3	**c_3**	c_8	c_1	c_1	c_2
2	c_{10}	c_9	c_6	c_2	c_7	c_4	c_6	c_4	c_5	c_{10}
3	c_6	c_{10}	**c_1**	c_5	c_9	c_5	c_4	c_2	**c_2**	**c_9**
4	$\emptyset$	c_6	c_7	c_8	**c_{10}**	c_1	c_2	c_6	c_{10}	c_4
5		c_3	c_4	c_1	c_1	c_9	c_9	c_3	$\emptyset$	c_1
6		c_7	c_8	c_7	c_5	c_7	$\emptyset$	**c_7**		c_7
7		c_8	c_3	c_3	c_4	c_{10}		c_5		c_5
8		c_1	c_9	c_9	c_2	c_8		c_{10}		c_6
9		c_5	c_{10}	c_{10}	c_6	c_2		c_8		c_3
10		c_2	c_2		c_8			c_9		c_8

We execute the student-proposing DA. Notice how truncations cause severe "immediate rejections" even when colleges are empty.

Round 1.
Students propose to their top choices.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
~~s_8~~	~~s_{10}~~	s_6	s_2	~~s_1~~	s_4		~~s_7~~		
~~s_9~~		~~s_5~~		~~s_3~~					

Explanation: Look closely at the rejections. College c_5 rejects *both* s_1 and s_3 immediately, because neither student is on c_5's acceptable list $(s_7, s_2, s_8, s_4, \emptyset)$. Similarly, c_1, c_2, c_8 reject their applicants purely because they fail to meet the individual rationality threshold.

Rejected: $s_1, s_3, s_5, s_7, s_8, s_9, s_{10}$.

Round 2.
The mass of rejected students propose to their second choices.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
		s_6	s_2	~~s_9~~	s_4	s_5			s_1
			~~s_8~~		~~s_3~~				~~s_{10}~~
					~~s_7~~				

Explanation: c_{10} receives s_1 and s_{10}; it prefers s_1. c_6 holds s_4 and rejects both s_3 and s_7. c_4 rejects s_8. c_5 continues to be empty because s_9 is unacceptable.
Rejected: $s_3, s_7, s_8, s_9, s_{10}$.

Round 3.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_9	s_6	s_2		s_4	s_5		s_{10}	s_1
	~~s_8~~		~~s_7~~						

Explanation: c_1 accepts s_3. c_9 accepts s_{10}. c_2 receives s_9 and s_8; s_9 is acceptable, but s_8 is unacceptable! Thus c_2 holds s_9 and rejects s_8.
Rejected: s_7, s_8.

Rounds 4 & 5.
$s_7 \to c_2$ (unacceptable, rejected). $s_8 \to c_6$ (rejected, $s_4 \succ s_8$).
Next: $s_7 \to c_9$ (rejected, $s_{10} \succ s_7$). $s_8 \to c_3$ (rejected, $s_6 \succ s_8$).
Rejected: s_7, s_8.

Round 6 (The Dropouts Begin).
Student s_7 has exhausted their list (reaching $\emptyset$ after c_9). s_7 **drops out of the market and remains permanently unmatched.**
Meanwhile, s_8 proposes to c_7.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_3	s_9	s_6	s_2		s_4	s_8		s_{10}	s_1
						~~s_5~~			

Explanation: c_7 prefers the new s_8 over the held s_5.
Rejected: s_5 (next c_9).

Rounds 7, 8, & 9.
$s_5 \to c_9$ (rejected, $s_{10} \succ s_5$).
$s_5 \to c_{10}$ (held, $s_5 \succ s_1$). s_1 is bumped!
$s_1 \to c_6$ (rejected, $s_4 \succ s_1$).

Round 10.
Student s_1 has exhausted their truncated list (reaching $\emptyset$ after c_6). s_1 **drops out of the market.** No further proposals can be made. The algorithm terminates.

The student-optimal stable matching μ_S under truncated preferences is:

College	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Student	s_3	s_9	s_6	s_2	$\emptyset$	s_4	s_8	$\emptyset$	s_{10}	s_5

Notice that students s_1 and s_7, as well as colleges c_5 and c_8, remain completely unmatched. This is the profound impact of the Individual Rationality constraint: stability now requires that some participants go home empty-handed rather than accept sub-par matches.

The introduction of truncated preferences is not just a way to model outside options (Individual Rationality); it also reveals a powerful method of strategic manipulation in matching markets.

If the mechanism asks participants to submit a rank-order list, can an agent benefit by *pretending* that certain acceptable partners are actually unacceptable? In other words, can an agent game the system by submitting an artificially shortened list? This question was famously answered by Alvin Roth.

Theorem 6.29: Strategic Truncation (Roth, 1982)

- **The Proposing Side:** In the college-proposing DA, no college can ever strictly improve its match by truncating its true preference list. (Symmetrically, no student can benefit from truncation in the student-proposing DA).
- **The Receiving Side:** In the student-proposing DA, a college **CAN** strictly improve its match by artificially truncating its preference list!

Remark (The Intuition Behind Strategic Truncation).

That the proposing side cannot benefit from truncation is a direct corollary of the Gale-Shapley Strategy-Proofness theorem: since truthful reporting is a dominant strategy for the proposers, no manipulation (including truncation) can help them.

But why does truncation work for the receiving side? Consider a college under the student-proposing DA. Suppose its true preference is $s_1 \succ_c s_2 \succ_c s_3$.

If the college reports truthfully, it might receive an application from s_2 early in the algorithm and tentatively hold them. Holding s_2 acts as a "safety net" for the college, but it also permanently removes s_2 from the broader applicant pool, effectively stabilizing the rest of the market.

If the college instead **truncates** its list to only include its top choice ($P_c' : s_1 \succ \emptyset$), it will ruthlessly reject s_2. This rejection forces s_2 to apply to their next-choice college, potentially triggering a massive **chain of rejections** across the market. If this chain reaction eventually bumps s_1 from whichever college was holding them, s_1 will fall down their own preference list and apply to c. By burning its safety net, the college forcefully destabilizes the market to shake loose its top choice!

Of course, truncation is a high-risk, high-reward strategy: if the rejection chain does not circle back to bump s_1, the college will end up completely empty-handed ($\emptyset$).

6.3.6 The Set of Stable Matchings

We now investigate the structure of the set of all stable matchings. The key results are: (i) the student-proposing DA produces the best stable matching for students, (ii) the interests of the two sides are diametrically opposed, and (iii) the set of stable matchings forms a lattice.

Example (A 4×4 Matching Problem).

Consider 4 students and 4 colleges with the following preferences:

Rank	c_1	c_2	c_3	c_4
1	s_1	s_2	s_3	s_4
2	s_2	s_1	s_4	s_3
3	s_3	s_4	s_1	s_2
4	s_4	s_3	s_2	s_1

Rank	s_1	s_2	s_3	s_4
1	c_4	c_3	c_2	c_1
2	c_3	c_4	c_1	c_2
3	c_2	c_1	c_4	c_3
4	c_1	c_2	c_3	c_4

Notice the "anti-diagonal" structure: colleges prefer students in order s_1, s_2, s_3, s_4, while students prefer colleges in the reverse order c_4, c_3, c_2, c_1.

Student-Proposing DA. Each student proposes to their top choice: $s_1 \to c_4$, $s_2 \to c_3$, $s_3 \to c_2$, $s_4 \to c_1$. Every college receives exactly one proposal. No rejections—the algorithm terminates immediately.

$$\mu_S : \quad (c_1, s_4),\ (c_2, s_3),\ (c_3, s_2),\ (c_4, s_1).$$

Every student gets their *first* choice. Every college gets their *last* choice.

College-Proposing DA. Each college proposes to their top choice: $c_1 \to s_1$, $c_2 \to s_2$, $c_3 \to s_3$, $c_4 \to s_4$. Every student receives exactly one proposal. No rejections.

$$\mu_C : \quad (c_1, s_1),\ (c_2, s_2),\ (c_3, s_3),\ (c_4, s_4).$$

Every college gets their *first* choice. Every student gets their *last* choice.

A third stable matching. Consider:

$$\bar{\mu} : \quad (c_1, s_3),\ (c_2, s_1),\ (c_3, s_4),\ (c_4, s_2).$$

One can verify that $\bar{\mu}$ has no blocking pair and is therefore stable, yet $\bar{\mu} \neq \mu_S$ and $\bar{\mu} \neq \mu_C$. In fact, this small market has **10 stable matchings** in total—the two DA solutions plus 8 others.

Student-Optimality of DA

Theorem 6.30: Student-Optimality of DA

The stable matching μ_S produced by the student-proposing DA is **weakly preferred by every student** to any other stable matching. That is, for every student s and every stable matching μ:

$$\mu_S^{-1}(s) \succsim_s \mu^{-1}(s).$$

By symmetry, the college-proposing DA produces the college-optimal stable matching μ_C.

Proof for Theorem

We first introduce a key definition.

Definition 6.31: Possible College

A college c is **possible** for student s if there exists some stable matching μ such that $\mu(s) = c$.

Claim

In the student-proposing DA, no student s is ever rejected by a college c that is possible for s.

Proof for Claim.

By induction on the number of steps in the algorithm.

Base case. At step 1, every student proposes to their top choice. No rejections have occurred yet, so the claim holds vacuously.

Inductive step. Suppose that up to step $t-1$, no student has been rejected by a college that is possible for them. Suppose at step t, student s is rejected by college c, because c prefers another student s' (whom it holds) to s:

$$s' \succ_c s.$$

We need to show that c is not possible for s.

Since s' is currently proposing to c at step t, student s' must have already been rejected by every college they prefer to c. By the induction hypothesis, all those colleges that rejected s' were not possible for s'. Therefore, c is the best possible college for s': in any stable matching μ, we have $c \succsim_{s'} \mu(s')$.

Now suppose, for contradiction, that c is possible for s. Then there exists a stable matching μ with $\mu(s) = c$. In this matching μ, student s' is matched to $\mu(s') \neq c$ (since s has c). But we just argued $c \succsim_{s'} \mu(s')$. Since $\mu(s') \neq c$ (because s occupies c in μ) and preferences are strict, this implies $c \succ_{s'} \mu(s')$. Moreover, c strictly prefers s' to $s = \mu(c)$. This means (c, s') blocks μ—contradicting the stability of μ. Therefore c is not possible for s.

The theorem follows immediately. Since no student is ever rejected by a possible college during the algorithm, each student's final match under μ_S is the *best* college that is achievable in any stable matching.

Opposition of Interests

The following proposition reveals that the two sides of the market have *diametrically opposed* interests over stable matchings.

Proposition 6.32: Opposition of Interests

If μ and μ' are both stable matchings and every student weakly prefers μ to μ', then every college weakly prefers μ' to μ.

Proof for Proposition.

Suppose all students weakly prefer μ to μ'. Suppose for contradiction that some college c strictly prefers μ to μ', i.e., $\mu(c) \succ_c \mu'(c)$. Let $s = \mu(c)$. Then:

- College c prefers s to $\mu'(c)$: $s \succ_c \mu'(c)$.
- Student s weakly prefers μ to μ', meaning $c = \mu^{-1}(s) \succsim_s \mu'^{-1}(s)$.

If s strictly prefers c to $\mu'^{-1}(s)$, then (c, s) blocks μ', contradicting stability. If s is indifferent, then since preferences are strict, $\mu^{-1}(s) = \mu'^{-1}(s) = c$, which implies $\mu(c) = \mu'(c) = s$, contradicting our assumption that c strictly prefers μ to μ'.

Corollary 6.33: Pessimality

The student-optimal stable matching μ_S is simultaneously the **college-pessimal** stable matching: every college gets its worst partner among all stable matchings. Symmetrically, μ_C is the student-pessimal stable matching.

The Lattice of Stable Matchings

Definition 6.34: Lattice of Stable Matchings

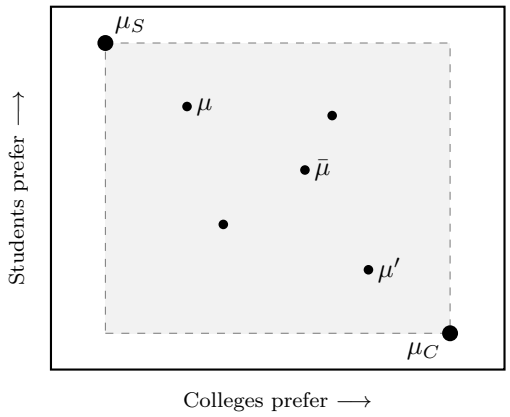
Let $\mathcal{M}$ denote the set of all stable matchings. Define a partial order $\geq_S$ on $\mathcal{M}$ from the students' perspective: $\mu \geq_S \mu'$ if every student weakly prefers μ to μ'. The set $(\mathcal{M}, \geq_S)$ forms a **distributive lattice**. This means:

- For any two stable matchings μ and μ', there exist stable matchings $\mu \vee \mu'$ (the *join*: student-best of the two) and $\mu \wedge \mu'$ (the *meet*: student-worst of the two).
- The lattice has a unique maximum element μ_S (student-optimal) and a unique minimum element μ_C (college-optimal, equivalently student-pessimal).

By the Opposition of Interests, the college ordering is exactly the reverse: what is the join for students is the meet for colleges, and vice versa.

Remark.

In the 4×4 example above, μ_S sits at the top of the lattice (every student's favorite) and μ_C at the bottom. All 8 remaining stable matchings lie strictly between these two extremes. The following diagram illustrates the structure: the vertical axis represents student welfare (higher is better for students) and the horizontal axis represents college welfare (further right is better for colleges). By the Opposition of Interests, μ_S occupies the upper-left corner and μ_C the lower-right corner. All other stable matchings are confined to the rectangular region spanned by these two extremes.



Moving toward μ_S makes every student weakly better off and every college weakly worse off. Moving toward μ_C does the opposite. Any stable matching must lie weakly within the bounds set by these two extremes.

The Rural Hospital Theorem

When we allow preferences to include truncations—i.e., agents may find some partners *unacceptable*, so that not every agent is necessarily matched—the following striking result holds.

Theorem 6.35: Rural Hospital Theorem (Lone Wolf Theorem)

If a student (or college) is unmatched in **some** stable matching, then that student (or college) is unmatched in **all** stable matchings.

Proof for Theorem

We prove the result for the case of one-to-one matching ($q_c = 1$ for all c).

Let $S_u \subseteq \mathcal{S}$ be the set of students unmatched in μ_S (the student-optimal stable matching), and let $C_u \subseteq \mathcal{C}$ be the set of colleges unmatched in μ_S. Similarly, let S'_u and C'_u be the sets of students and colleges unmatched in μ_C (the college-optimal stable matching).

Step 1: $S_u \subseteq S'_u$. Since μ_S is the student-optimal stable matching, every student gets their best possible partner across all stable matchings. If a student s is unmatched even in μ_S, then s cannot be matched in any stable matching. In particular, s is unmatched in μ_C. Hence $S_u \subseteq S'_u$.

Step 2: $C'_u \subseteq C_u$. By symmetric reasoning, μ_C is the college-optimal stable matching, so every college gets its best possible partner. If a college c is unmatched in μ_C, then c is unmatched in every stable matching, including μ_S. Hence $C'_u \subseteq C_u$.

Step 3: Counting argument. In any one-to-one matching, the number of matched students equals the number of matched colleges. Therefore:

$$|\mathcal{S}| - |S_u| = |\mathcal{C}| - |C_u| \qquad \text{and} \qquad |\mathcal{S}| - |S'_u| = |\mathcal{C}| - |C'_u|.$$

From Step 1, $|S_u| \leq |S'_u|$, so the number of matched students in μ_S is at least the number in μ_C. From Step 2, $|C'_u| \leq |C_u|$, so the number of matched colleges in μ_C is at least the number in μ_S. But matched students = matched colleges in each matching, so:

$$\underbrace{|\mathcal{S}| - |S_u|}_{\text{matched in } \mu_S} \geq \underbrace{|\mathcal{S}| - |S'_u|}_{\text{matched in } \mu_C} = \underbrace{|\mathcal{C}| - |C'_u|}_{\text{matched in } \mu_C} \geq \underbrace{|\mathcal{C}| - |C_u|}_{\text{matched in } \mu_S}.$$

The leftmost and rightmost terms are equal (both count matched pairs in μ_S), so all inequalities are equalities. This forces $|S_u| = |S'_u|$ and $|C_u| = |C'_u|$. Combined with $S_u \subseteq S'_u$ and $C'_u \subseteq C_u$, we conclude:

$$S_u = S'_u \qquad \text{and} \qquad C_u = C'_u.$$

Since every stable matching μ satisfies $S_u \subseteq S_u^\mu \subseteq S'_u$ (the first inclusion by student-optimality of μ_S, the second by college-optimality of μ_C), and $S_u = S'_u$, we have $S_u^\mu = S_u$ for all stable matchings μ.

Remark.

The name "Rural Hospital" comes from the practical observation in medical residency matching: hospitals in rural areas that fail to fill all their positions under one stable matching will fail to fill them under *every* stable matching. Switching from the hospital-proposing to the resident-proposing DA (or any other stable mechanism) cannot help a rural hospital that is fundamentally undesirable to residents. The set of matched agents is invariant across stable matchings—only the *partners* can change, not whether one is matched at all.

6.4 The Boston Mechanism (Immediate Acceptance)

To truly appreciate the elegance and power of the Deferred Acceptance (DA) algorithm, it is instructive to study a mechanism that attempts to solve the same problem but fails spectacularly in its theoretical properties. The most famous example is the **Boston Mechanism**, historically used by the Boston Public School system (and many other school districts worldwide) before economists pointed out its severe flaws.

The Boston Mechanism is also known as the **Immediate Acceptance** mechanism. The critical difference from DA lies in the word "immediate": once a college accepts a student, that decision is permanent. Seats are locked in, and future applicants cannot displace already-admitted students, regardless of how highly the college ranks the new applicants.

Definition 6.36: The Boston Mechanism

1. **Round 1:** Each student proposes to their first-choice college. Each college looks at its proposals, accepts its most preferred students up to its quota, and rejects the rest. **These acceptances are final.** The college's quota is reduced accordingly.
2. **Round k ($k \geq 2$):** Each student rejected in Round $k-1$ proposes to their k-th choice college. Each college looks at its *remaining* unfilled seats (if any). It accepts its most preferred students among the new applicants up to its remaining quota, and rejects the rest. **These acceptances are final.**
3. **Repeat** until all students are matched or all available seats are filled.

Example (Boston Mechanism on the 10x10 Matching Problem).

Let us apply the Boston Mechanism to the exact same 10-student, 10-college problem. All colleges have quota $q = 1$.

College preferences (Boston Mechanism final matches highlighted):

Rank	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
1	s_1	s_1	s_{10}	s_6	s_7	s_4	s_1	s_6	s_8	s_6
2	s_2	s_6	s_2	s_1	s_2	s_8	s_4	s_4	s_{10}	s_5
3	s_3	s_9	s_4	s_5	s_8	s_2	s_3	s_2	s_6	$\mathbf{s_1}$
4	s_4	s_2	$\mathbf{s_6}$	$\mathbf{s_2}$	s_4	s_6	s_8	s_{10}	$\mathbf{s_9}$	s_4
5	s_5	s_5	s_8	s_7	s_{10}	s_{10}	$\mathbf{s_5}$	s_8	s_5	s_{10}
6	s_6	s_8	s_3	s_3	s_5	s_3	s_2	s_3	s_7	s_8
7	s_7	s_3	s_5	s_8	s_9	s_7	s_9	s_1	s_4	s_9
8	$\mathbf{s_8}$	s_4	s_7	s_4	s_6	s_1	s_6	s_9	s_3	s_2
9	s_9	s_7	s_9	s_{10}	$\mathbf{s_3}$	s_5	s_7	$\mathbf{s_7}$	s_1	s_3
10	s_{10}	$\mathbf{s_{10}}$	s_1	s_9	s_1	s_9	s_{10}	s_5	s_2	s_7

Student preferences (Boston Mechanism final matches highlighted):

Rank	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
1	c_5	$\mathbf{c_4}$	$\mathbf{c_5}$	$\mathbf{c_6}$	c_3	$\mathbf{c_3}$	$\mathbf{c_8}$	$\mathbf{c_1}$	c_1	$\mathbf{c_2}$
2	$\mathbf{c_{10}}$	c_9	c_6	c_2	$\mathbf{c_7}$	c_4	c_6	c_4	c_5	c_{10}
3	c_6	c_{10}	c_1	c_5	c_9	c_5	c_4	c_2	c_2	c_9
4	c_4	c_6	c_7	c_8	c_{10}	c_1	c_2	c_6	c_{10}	c_4
5	c_7	c_3	c_4	c_1	c_1	c_9	c_9	c_3	c_3	c_1
6	c_2	c_7	c_8	c_7	c_5	c_7	c_7	c_7	c_8	c_7
7	c_8	c_1	c_9	c_3	c_4	c_{10}	c_5	c_5	c_6	c_5
8	c_9	c_5	c_{10}	c_9	c_2	c_8	c_{10}	c_{10}	c_7	c_6
9	c_1	c_2	c_2	c_{10}	c_6	c_2	c_1	c_8	c_4	c_3
10	c_3	c_8	c_3	c_4	c_8	c_6	c_3	c_9	$\mathbf{c_9}$	c_8

We use the same chalkboard-style simulation. However, green text now represents a **permanent acceptance** (**admitted**), rather than a tentative hold. Once a column is filled with a green student, that college is closed forever.

Round 1.
All 10 students propose to their first-choice colleges.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s_8	**s_{10}**	**s_6**	**s_2**	**s_3**	**s_4**		**s_7**		
~~s_9~~		~~s_5~~		~~s_1~~					

Explanation: Colleges c_2, c_4, c_6, c_8 receive exactly one applicant and admit them immediately. Colleges c_1, c_3, c_5 face competition. They admit their preferred applicant (s_8, s_6, s_3 respectively) and reject the rest.
Status: Colleges $c_1, c_2, c_3, c_4, c_5, c_6, c_8$ are now **permanently closed**.
Rejected: s_9 (next c_5), s_5 (next c_7), s_1 (next c_{10}).

Round 2.
The three rejected students propose to their second-choice colleges.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
				FULL		**s_5**			**s_1**
				~~s_9~~					

Explanation: Students s_1 and s_5 apply to c_{10} and c_7, which are still empty. They are admitted permanently. Student s_9 applies to c_5. However, c_5 was already filled by s_3 in Round 1! Even though c_5 prefers s_9 (#7) over s_3 (#9), it is too late. The seat is gone. s_9 is automatically rejected.
Status: Colleges c_7, c_{10} are now **permanently closed**. Only c_9 remains open.
Rejected: s_9 (next c_2).

Rounds 3 through 9 (The Free-Fall of s_9).
Student s_9 has suffered a "cascading failure." Because he missed his first choice, he is now applying to colleges that were filled in Round 1.

- **Round 3:** $s_9 \to c_2$. **FULL** (filled by s_{10}). Rejected.
- **Round 4:** $s_9 \to c_{10}$. **FULL** (filled by s_1). Rejected.
- **Round 5:** $s_9 \to c_3$. **FULL** (filled by s_6). Rejected.
- **Round 6:** $s_9 \to c_8$. **FULL** (filled by s_7). Rejected.
- **Round 7:** $s_9 \to c_6$. **FULL** (filled by s_4). Rejected.
- **Round 8:** $s_9 \to c_7$. **FULL** (filled by s_5). Rejected.
- **Round 9:** $s_9 \to c_4$. **FULL** (filled by s_2). Rejected.

Round 10.
Student s_9 finally reaches his 10th and absolute worst choice, c_9. Luckily, c_9 is still empty.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
								s_9	

The algorithm terminates.

The **Boston Mechanism matching** μ_B is:

College	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Student	s_8	s_{10}	s_6	s_2	s_3	s_4	s_5	s_7	s_9	s_1

6.4.1 Properties of the Boston Mechanism

The simulation above provides everything we need to expose the theoretical failures of the Boston Mechanism. We examine the two most important properties: Pairwise Stability and Strategy-Proofness.

Theorem 6.37: The Boston Mechanism is NOT Stable

The allocation produced by the Boston Mechanism may contain blocking pairs, meaning it fails to guarantee a pairwise stable matching.

Proof for Theorem

We can prove this by simply identifying a blocking pair in the outcome μ_B of our example.

Under μ_B, student s_9 is matched with c_9 (his 10th choice), and college c_5 is matched with s_3 (its 9th choice). Now look at their true preferences:

- Student s_9 ranks c_5 as his 2nd choice, meaning $c_5 \succ_{s_9} \mu_B(s_9) = c_9$.
- College c_5 ranks s_9 as its 7th choice, meaning $s_9 \succ_{c_5} \mu_B(c_5) = s_3$.

Both s_9 and c_5 would strictly prefer to be matched with each other rather than their current assignments. Thus, (c_5, s_9) **forms a blocking pair**.

The instability arises precisely because of the "immediate acceptance" rule: c_5 was forced to permanently commit to s_3 in Round 1, preventing it from accepting the much stronger applicant s_9 who arrived in Round 2.

Theorem 6.38: The Boston Mechanism is NOT Strategy-Proof

Under the Boston Mechanism, truthful reporting is not a dominant strategy. Students can often obtain a strictly better outcome by misreporting their preferences.

Proof for Theorem

Because missing out on your first choice is heavily penalized (the "free-fall" effect seen with s_9), the Boston Mechanism forces students to play a high-stakes psychological game. Instead of ranking their true favorite first, students are incentivized to rank a "safe" school first to secure a seat in Round 1.

We demonstrate this with student s_9 from our example. Under truthful reporting,

s_9 listed c_1 as his 1st choice and c_5 as his 2nd choice. Because he lost the tie-breaker at c_1 in Round 1, he free-fell all the way to c_9, his worst possible outcome.

Suppose s_9 anticipates this and decides to deviate. He submits a fake preference list P'_{s_9} where he promotes his 2nd choice (c_5) to his 1st choice, abandoning the highly competitive c_1. The rest of the students report truthfully.

Modified Student Preferences (with s_9's misreport highlighted in blue):

Rank	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	$\mathbf{s'_9}$	s_{10}
1	c_5	c_4	c_5	c_6	c_3	c_3	c_8	c_1	$\mathbf{c_5}$	c_2
2	c_{10}	c_9	c_6	c_2	c_7	c_4	c_6	c_4	$\mathbf{c_1}$	c_{10}
3	c_6	c_{10}	c_1	c_5	c_9	c_5	c_4	c_2	c_2	c_9
4	c_4	c_6	c_7	c_8	c_{10}	c_1	c_2	c_6	c_{10}	c_4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Let us trace **Round 1** of the Boston Mechanism under this new profile. All students apply to their reported 1st choices.

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
$\mathbf{s_8}$	$\mathbf{s_{10}}$	$\mathbf{s_6}$	$\mathbf{s_2}$	$\mathbf{s_9}$	$\mathbf{s_4}$		$\mathbf{s_7}$		
		~~s_5~~		~~s_3~~					
				~~s_1~~					

Explanation of c_5's decision: College c_5 now receives Round 1 proposals from s_1, s_3, and the strategically misreporting s_9. Looking at c_5's true preference list: s_9 (#7) $\succ_{c_5}$ s_3 (#9) $\succ_{c_5}$ s_1 (#10). Because c_5 evaluates all three applicants simultaneously in Round 1, it permanently admits its favorite among them: s_9.

By lying and placing c_5 first, student s_9 completely avoided the free-fall and secured his true 2nd choice (c_5). This outcome is strictly better for him than the 10th choice (c_9) he received when telling the truth. Since a profitable deviation exists, the Boston mechanism is **not strategy-proof**.

Remark (The Tragedy of the Boston Mechanism).

The Boston Mechanism systematically rewards sophisticated, strategic families while punishing honest, naive ones. A naive student who truthfully applies to a competitive "reach" school wastes their critical Round 1 proposal. By the time they are rejected and move to their "match" schools in Round 2, those schools have already been filled by strategic students who ranked them first. This profound inequity is why many school districts have replaced it with the Strategy-Proof Deferred Acceptance algorithm.

Remark (Chapter Summary).

Matching theory studies allocation problems in which prices cannot do all the work—either because money is unavailable (school choice, kidney exchange, public housing)

or because preferences are intrinsically two-sided (medical residents and hospitals, men and women). Two foundational algorithms organize the field. *Top trading cycles* (TTC) solves the one-sided housing-market problem: starting from arbitrary endowments, it produces the unique core allocation, is individually rational, Pareto efficient, and group strategy-proof. *Deferred acceptance* (Gale-Shapley DA) solves the two-sided market: it produces a stable matching, is strategy-proof for the proposing side, and the proposing side gets its optimal stable matching. Three trade-offs run through the chapter. *Stability vs. optimality*: stability picks out a lattice of matchings; the proposing side gets the top, the receiving side gets the bottom. *Strategy-proofness vs. efficiency*: in priority-based settings, DA is strategy-proof but not Pareto efficient; TTC is Pareto efficient but ignores priorities. *Truthfulness vs. welfare*: the Boston mechanism rewards strategic sophistication and harms honest participants, motivating the move to DA in real-world school choice systems (NYC 2003, Boston 2005).

Part V

Information and Dynamic Games

Part VI

Problem Sets and Solutions

Part VII

Exams and Solutions