1. Stable-matching

1.1 Description

Design a self-reinforcing admission process.

Hospital h and student s form an **unstable pair** if both:
• h prefers s to one of its admitted students.
• s prefers h to assigned hospital.

Stable assignment: assignment with no unstable pairs

Input. A set of n hospitals H and a set of n students S.
• Each hospital h ∈ H ranks students.
• Each student s ∈ S ranks hospitals.

Def. A matching M is perfect if | M | = | H | = | S | = n.

	1st	2nd	3rd
Atlanta	Xavier	Yolanda	Zeus
Boston	Yolanda	Xavier	Zeus
Chicago	Xavier	Yolanda	Zeus

	1st	2nd	3rd
Xavier	Boston	Atlanta	Chicago
Yolanda	Atlanta	Boston	Chicago
Zeus	Atlanta	Boston	Chicago

Def. unstable pair if both: **(improvable by joint action)**

- h prefers s to matched student.
- s prefers h to matched hospital.

Stable matching problem. Given the preference lists of n hospitals and n students, find a stable matching (if one exists).

```
GALE–SHAPLEY (preference lists for hospitals and students) (O(n^2))
________________________________________________________________________________
INITIALIZE M to empty matching.
WHILE (some hospital h is unmatched and hasn’t proposed to every student)
  s ← first student on h’s list to whom h has not yet proposed.
  IF (s is unmatched)
    Add h–s to matching M.
  ELSE IF (s prefers h to current partner h')
    Replace h'–s with h–s in matching M.
  ELSE
    s rejects h.
RETURN stable matching M.
```

Gale–Shapley terminate, is perfect and stable and at the end produces a matching for every entity.

1.1.1 Efficient implementation

- Maintain a list of free hospitals (in a stack or queue).
- Maintain two arrays student[h] and hospital[s].
    - if h matched to s, then student[h] = s and hospital[s] = h
    - use value 0 to designate that hospital or student is unmatched

• For each hospital, maintain a list of students, ordered by preference.
• For each hospital, maintain a pointer to students in list for next proposal.

Students rejecting/accepting.

pref[]	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}
	8	3	7	1	4	5	6	2

inverse[]	1	2	3	4	5	6	7	8
	4^{th}	8^{th}	2^{nd}	5^{th}	6^{th}	7^{th}	3^{rd}	1^{st}

```
for i = 1 to n
    inverse[pref[i]] = i
```

Def. Student s is a valid partner for hospital h if there exists any stable matching in which h and s are matched.

Claim. All executions of Gale–Shapley yield hospital-optimal assignment.

Corollary. Hospital-optimal assignment is a stable matching!

Claim. Gale–Shapley matching S* is hospital-optimal.

Student-pessimal assignment. Each student receives worst valid partner.

Claim. Gale–Shapley finds student-pessimal stable matching M*.

1.1.2 Extensions

Extension 1. Some participants declare others as unacceptable.
Extension 2. Some hospitals have more than one position.
Extension 3. Unequal number of positions and students.

Def. Matching M is **unstable** if there is a hospital h and student s such that:
• h and s are acceptable to each other; and
• Either s is unmatched, or s prefers h to assigned hospital; and
• Either h does not have all its places filled, or h prefers s to at least one of its assigned students.

1.2 Representative problems

Interval scheduling: find maximum cardinality subset of mutually compatible jobs with start and finished times given.

Bipartite matching: Given a bipartite graph G = (L U R, E),find a max cardinality matching.
Def. A subset of edges $M \subseteq E$ is a **matching** if each node appears in exactly one edge in M.

Independent set: A subset $S \subseteq V$ is independent if for every $(u, v) \in E$, either $u \notin S$ or $v \notin S$ (or both).

Competitive facility location: Input. Graph with weight on each node.
Game. Two competing players alternate in selecting nodes. Not allowed to select a node if any of its neighbours have been selected.
Goal. Select a maximum weight subset of nodes.