5112

Algorithms

website

Slack

Gale-Shapley Algorithm (1962)

while ∃ an unmatched university U:
  C ← next candidate
  if C prefers U to their current match:
    match C to U.
  move next candidate pointer

Preferences

Alex	Jennifer	Raj
C	R	R
R	C	C
P	P	P

Cornell	Princeton	Rutgers
J	A	J
→ A	→ J	R
R	R	A

Alex —— Cornell
Jennifer —— Princeton
Raj —— Rutgers

Can we run out of candidates to make offers to?

Invariant: Once a candidate has an offer, they will always have an offer.

Gale-Shapley Algorithm (1962)

while ∃ an unmatched university U:
  C ← next candidate
  if C prefers U to their current match:
    match C to U.
  move next candidate pointer

Preferences

Alex	Jennifer	Raj
C	R	R
R	C	C
P	P	P

Cornell	Princeton	Rutgers
J	A	J
-> A	-> J	R
R	R	A

Alex — Cornell
Jennifer — Princeton
Raj — Rutgers

Can we run out of candidates to make offers to?

Invariant: Once a candidate has an offer, they will always have an offer.

Cornell:
Make an offer to Jennifer

> 1 pair " " " "

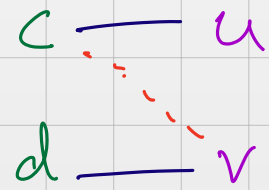
> 2 pair Alex

" " " " Raj.

Cannot be matched
⇒ will accept

The algorithm terminates with a matching

Is this matching stable?



Observation: Candidates matches only improve.

Let c, v be an unmatched pair.
Let u be c's match and d be v's match

Suppose v prefers c to d, then v made an offer to c.
Case 1: c rejected $\Rightarrow$ c preferred another offer.
$\Rightarrow$ c prefers u to v.
$\Rightarrow$ not unstable

Is this matching stable?

Observation: Candidates matches only improve.

Let c, v be an unmatched pair.
Let u be c's match and d be v's match

Suppose v prefers c to d, then v made an offer to c.
Case 2: c accepted: but c eventually accepts an offer from u.

not unstable $\Rightarrow$ c prefers u by the observation.

Gale-Shapley Algorithm (1962)

while $\exists$ an unmatched university U:
  $C \leftarrow$ next candidate
  if C prefers U to their current match:
    match C to U.
  move next candidate pointer

How long does it take?

Can do at most n^2 loop iterations

How long does a loop iteration take?

Store unmatched universities in a list
$O(1)$ time

Store university prefs in arrays.

	Cornell	Princeton	Rutgers
Alex	1	3	2
Jennifer	2	3	1
Raj	2	3	1

Can look up cand. prefs in $O(1)$ time.

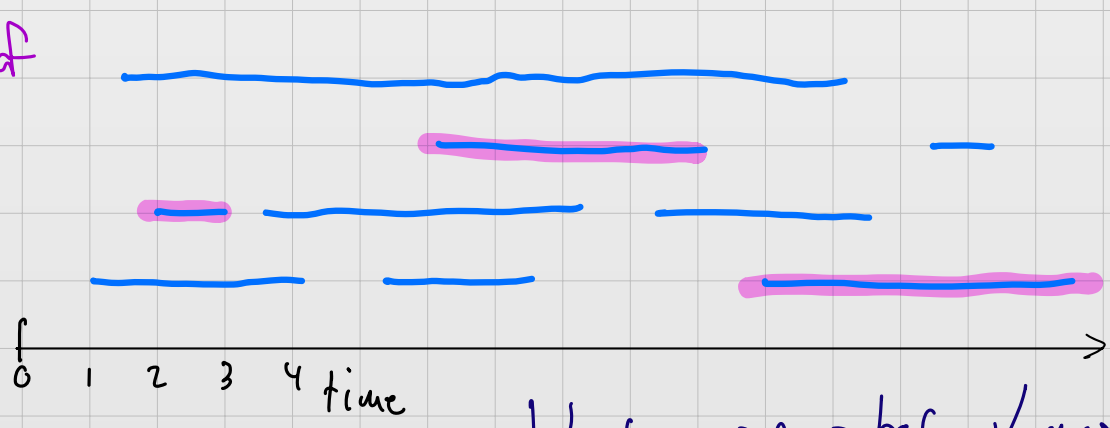
$\Rightarrow$ Each loop is $O(1)$ time

$\Rightarrow$ $O(n^2)$ running time.

Interval Scheduling

Input: Jobs $J_1, J_2, \ldots, J_n$ $\quad J_i = (s_i, f_i)$

A subset of jobs are compatible if no two overlap



Want a comp. subset w/ most # of jobs

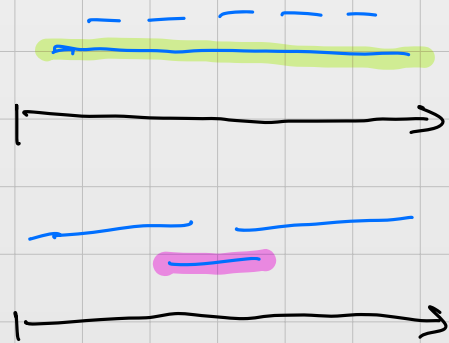
Greedy Algorithms

A greedy algorithm proceeds in steps, making a locally optimal decision at each step, without ever looking into the future.

Interval Scheduling

Greedy Algorithms

1. Earliest start time (FCFS)
   (not maximal)



2. Shortest job
   (not maximal)

3. Fewest conflicts (not maximal)

4. Earliest finish time (looks promising)

Intuition: Choose the job that frees the resource as soon as possible