CSE 421

Greedy Algorithms / Minimizing Lateness

Yin Tat Lee

Homework 1

- Don’t worry if your homework 1 grade is not ideal.
- You can recover score by extra credit.
- See the website for updated late and regrade policy.

For Question 1:

1. If m is the first man on the woman w’s preference list and w is the first woman on m’s preference list, does it have to be the case that m and w must be matched to each other in every stable matching?

Some students prove the statement for the matching by G-S.

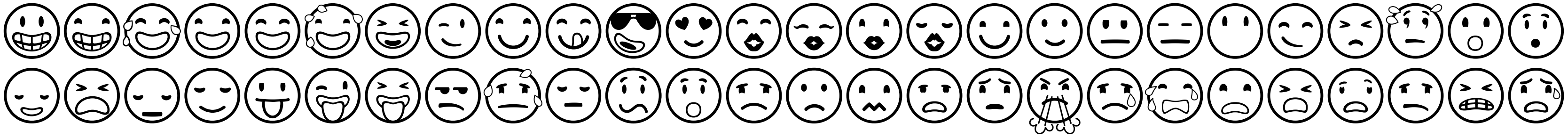
This is wrong because there are stable matchings not produced by G-S.

Homework 1

If the solution involves reducing the problem into a question we solved in class, then the solution should look like:

<u>Reduction Outline</u>

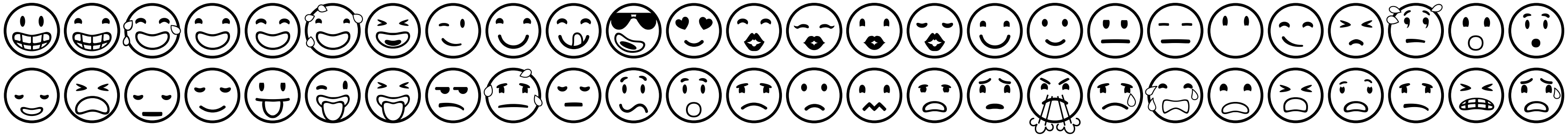
Easy for TA to grade



<u>Proof of Runtime/Termination</u>

For Q2, involves proving stable matching satisfies (*)

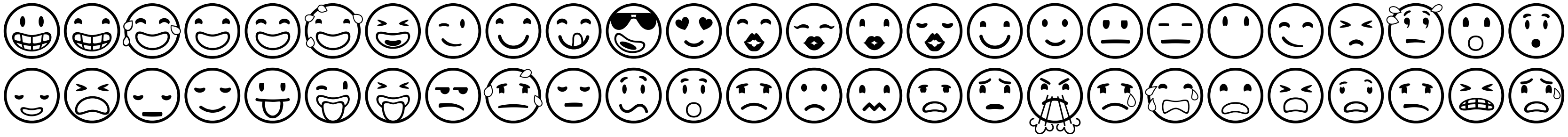
<u>Proof of Correctness</u>



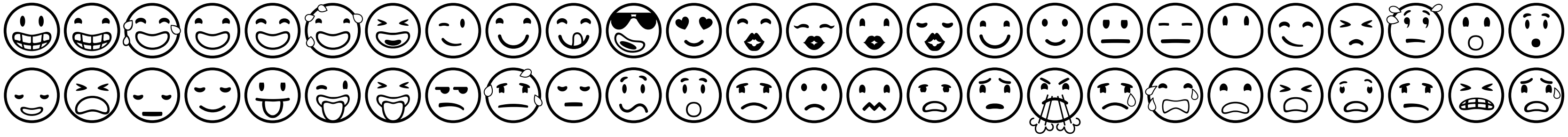
Homework 1

If the solution involves a new algorithm (usually modified from an algo in class), then the solution should look like:

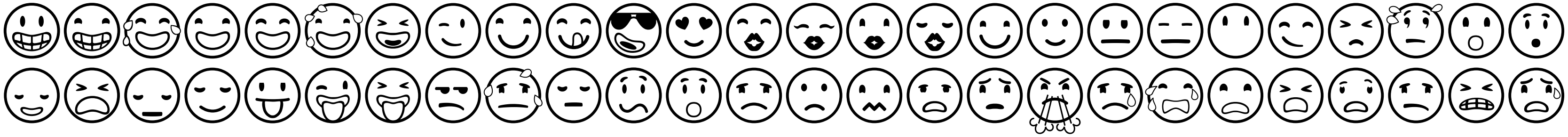
<u>Algorithm Outline</u>



<u>Proof of Runtime/Termination</u>



<u>Proof of Correctness</u>



Greedy Analysis Strategies

Greedy algorithm stays ahead

- Show that after each step of the greedy algorithm, its solution is at least as good as any other algorithm's.

Structural

- Discover a simple "structural" bound asserting that every possible solution must have a certain value. Then show that your algorithm always achieves this bound.

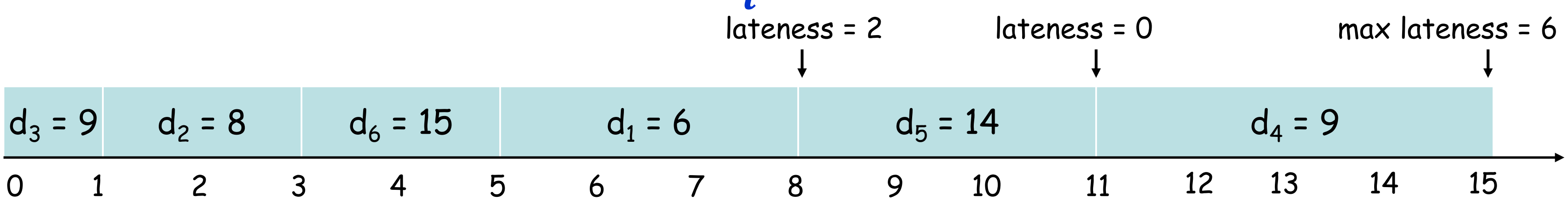
Exchange argument

- Gradually transform any solution to the one found by the greedy algorithm without hurting its quality.

Scheduling to Minimizing Lateness

- Similar to interval scheduling.
- Instead of start and finish times, request $\boldsymbol{i}$ has
  - ➢ Time Requirement $\boldsymbol{t_i}$ which must be scheduled in a contiguous block
  - ➢ Deadline $\boldsymbol{d_i}$ by which time the request would like to be finished
- Requests are scheduled into time intervals $[\boldsymbol{s_i}, \boldsymbol{f_i}]$ s.t. $\boldsymbol{t_i = f_i - s_i}$.
- Lateness for request $\boldsymbol{i}$ is
  - If $\boldsymbol{d_i < f_i}$ then request $\boldsymbol{i}$ is late by $\boldsymbol{L_i = f_i - d_i}$ otherwise its lateness $\boldsymbol{L_i = 0}$
- **Goal:** Find a schedule that minimize the Maximum lateness $\boldsymbol{L = \max_i L_i}$

	1	2	3	4	5	6
t_j	3	2	1	4	3	2
d_j	6	8	9	9	14	15



Minimizing Lateness: Greedy Algorithms

Greedy template. Consider jobs in some order.

- [Shortest processing time first]
  Consider jobs in ascending order of processing time t_j.

	1	2
t_j	1	10
d_j	100	10

counterexample

- [Smallest slack]
  Consider jobs in ascending order of slack $d_j - t_j$.

	1	2
t_j	1	10
d_j	2	10

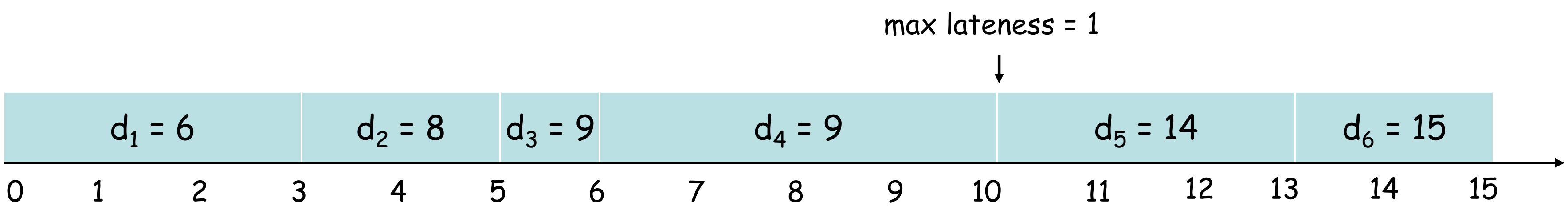
counterexample

- [Earliest deadline first]
  Consider jobs in ascending order of deadline d_j.

Greedy Algorithm: Earliest Deadline First

Sort deadlines in increasing order $(d_1 \le d_2 \le \cdots \le d_n)$
$f \leftarrow 0$
for $i \leftarrow 1$ to n to
$\quad s_i \leftarrow f$
$\quad f_i \leftarrow s_i + t_i$
$\quad f \leftarrow f_i$
end for

	1	2	3	4	5	6
t_j	3	2	1	4	3	2
d_j	6	8	9	9	14	15



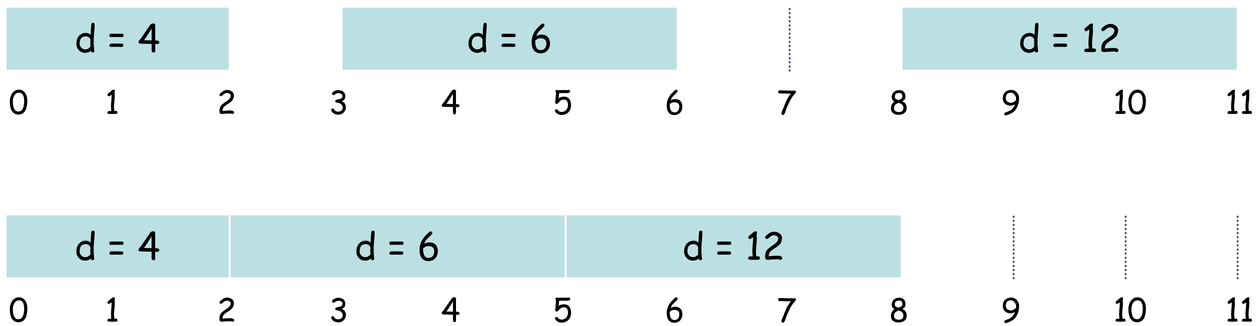
Proof for Greedy Algorithm: Exchange Argument

- We will show that if there is another schedule $\boldsymbol{O}$ (think optimal schedule) then we can gradually change $\boldsymbol{O}$ so that
  - at each step the maximum lateness in $\boldsymbol{O}$ never gets worse
  - it eventually becomes the same cost as $\boldsymbol{A}$

Minimizing Lateness: No Idle Time

Observation.

- There exists an optimal schedule with no idle time.



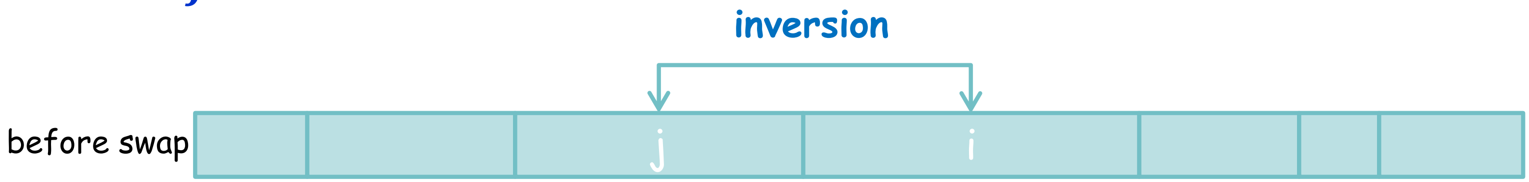
Observation.

- The greedy schedule has no idle time.

Minimizing Lateness: Inversions

Definition

- An inversion in schedule S is a pair of jobs i and j such that $d_i < d_j$ but j scheduled before i.



Observation

- Greedy schedule has no inversions.

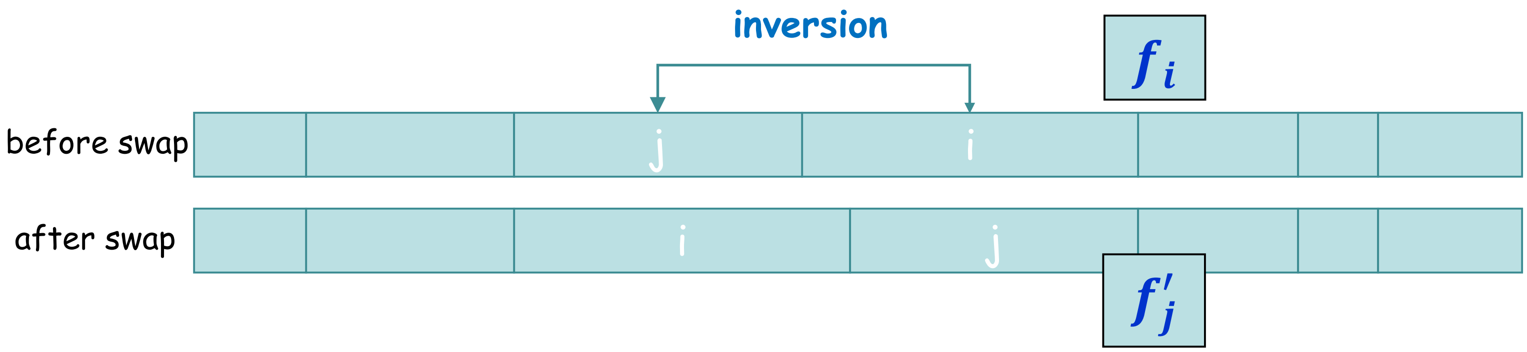
Observation

- If a schedule (with no idle time) has an inversion, it has one with a pair of inverted jobs scheduled **consecutively**.
  - Why? If all jobs are in order consecutively, then no inversion.

Minimizing Lateness: Inversions

Definition

- An inversion in schedule S is a pair of jobs i and j such that $d_i < d_j$ but j scheduled before i.

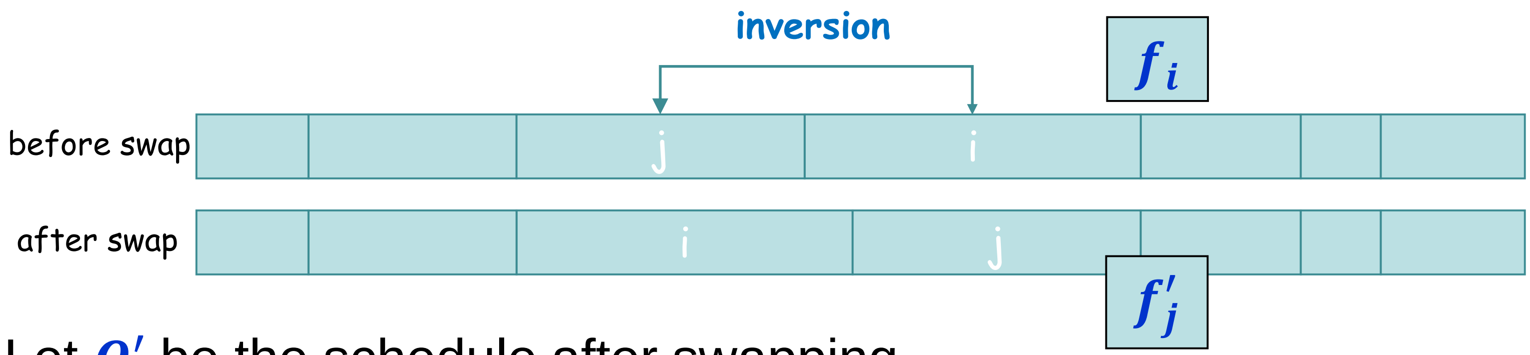


Claim

- Swapping two adjacent, inverted jobs reduces the number of inversions by one and does not increase the max lateness.

Minimizing Lateness: Inversions

Lemma: Swapping two adjacent, inverted jobs does not increase the maximum lateness.



Proof: Let O' be the schedule after swapping.

- Lateness $L_i' \leq L_i$ since i is scheduled earlier in O' than in O
- Requests i and j together occupy the same total time slot in both schedules
  - All other requests $k \neq i, j$ have $L_k' = L_k$
  - $f_j' = f_i$ so $L'_j = f'_j - d_j = f_i - d_j < f_i - d_i = L_i$
- Maximum lateness has not increased!

Optimal schedules and inversions

Did we finish the proof for greedy?

Claim: There is an optimal schedule with no idle time and no inversions

Proof:

- By previous argument there is an optimal schedule $\boldsymbol{O}$ with no idle time
- If $\boldsymbol{O}$ has an inversion then it has a **consecutive** pair of requests in its schedule that are inverted and can be swapped without increasing lateness
- Eventually these swaps will produce an optimal schedule with no inversions
  - Each swap decreases the number of inversions by **1**
  - There are at most $\boldsymbol{n(n-1)/2}$ inversions. (we only care that this is finite.)

Idleness and Inversions are the only issue

Claim: All schedules with no inversions and no idle time have the same maximum lateness

Proof:

- Schedules can differ only in how they order requests with equal deadlines
- Consider all requests having some common deadline d
- Maximum lateness of these jobs is based only on the finish time of the last of these jobs but the set of these requests occupies the same time segment in both schedules
  - Last of these requests finishes at the same time in any such schedule.

Earliest Deadline First is optimal

We know that

- There is an optimal schedule with no idle time or inversions
- All schedules with no idle time or inversions have the same maximum lateness
- EDF produces a schedule with no idle time or inversions

Therefore

- EDF produces an optimal schedule

Life Wisdom:

- Finish your jobs according to deadline!
- ☹ Unfortunately, we don't see all jobs when born.