TODAY: Greedy algorithms
- Job sched
- Classsched
- Gale-Shapley
- change making

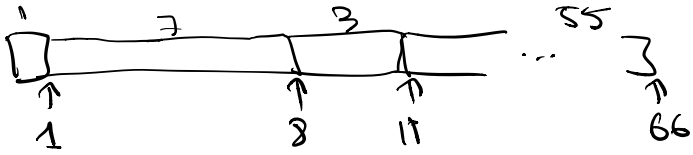
Restaurant $o_1, \ldots, o_n$ orders

$\ell(1) \ldots$ $\ell(n)$ → how long each order takes to cook

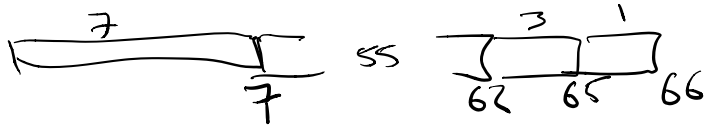
tip → subtract \$1 for each minute waiting

minimize the total waiting time

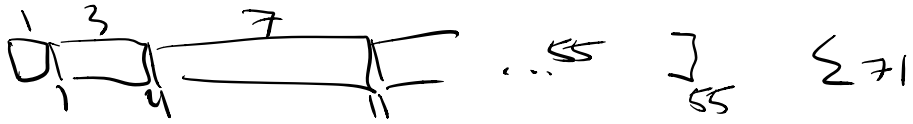
lengths 1 7 3 55



≤ 86



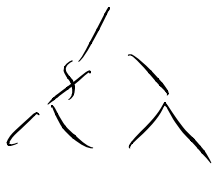
≤ 200



≤ 71

Shortest Job First

while jobs left
schedule shortest job



Suppose greedy schedule $g_1, \ldots, g_n$ $\ell(g_i) \leq \ell(g_{i+1})$

Optimal schedule $o_1, \ldots, o_n$ o_k o_{k+1} such that $g(o_k) > g(o_{k+1})$

swap o_k o_{k+1}
$o'_k = o_{k+1}$ $o'_{k+1} = o_k$

$w(o'_k) = w(o_{k+1}) - \ell(o_k)$

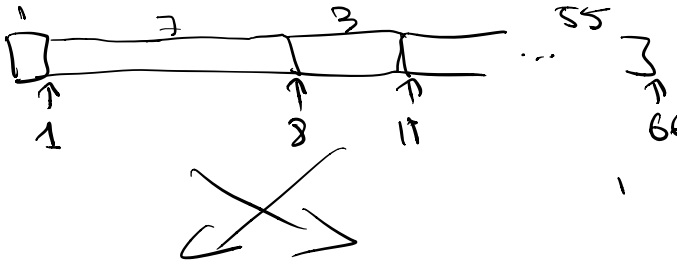
$w(o'_{k+1}) = w(o_k) + \ell(o_{k+1})$

$\leq w(o'_k) = \leq (w(o_k) + \ell(o_{k+1}) - \ell(o_k))$ similar bisser

$g(O_{k+1}) < g(O_k)$ $\leq \leq w(O_k)$

$\ell(O_{k+1}) \leq \ell(O_k)$

We can produce optimal schedule in greedy order



≤ 86

11 − 7 8 + 3

Person	tip		wait	
1	−1$	/ min	1	1/1
2	−5$	/ min	7	7/5
3	−2$	/ min	3	3/2
4	−6$	/ min	55	55/6

order by $\ell(i) / tip(i)$

Shortest Job First

while jobs left
schedule shortest job

v1 $O(n^2)$

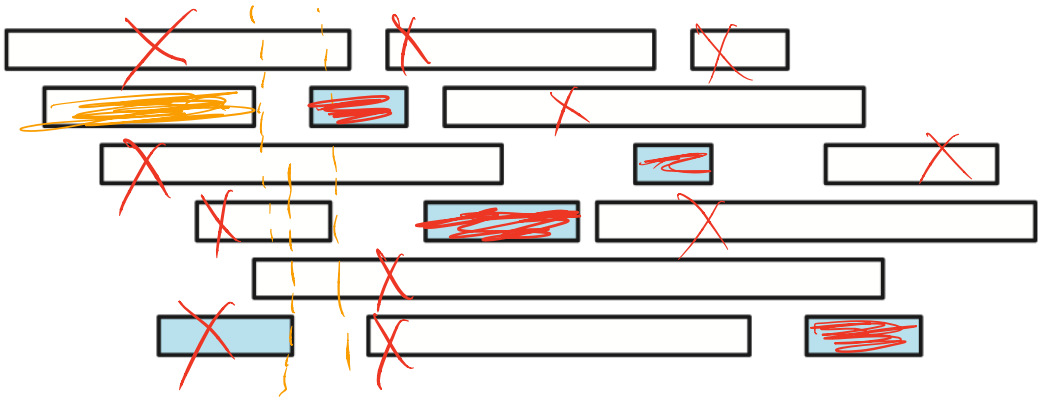
```
sched(jobs):
    result = []
    while (len(jobs) > 0
        j = next(jobs)          O(n)
        result.append(j)        O(1)
        jobs.remove(j)          O(n)?
```

n iterations

v2 $O(n \log n)$

```
for j in jobs
    → insert(j, q)              log n
    while q not empty
    → extract min(q)            log n
```

v3 return sorted(jobs) $O(n \log n)$



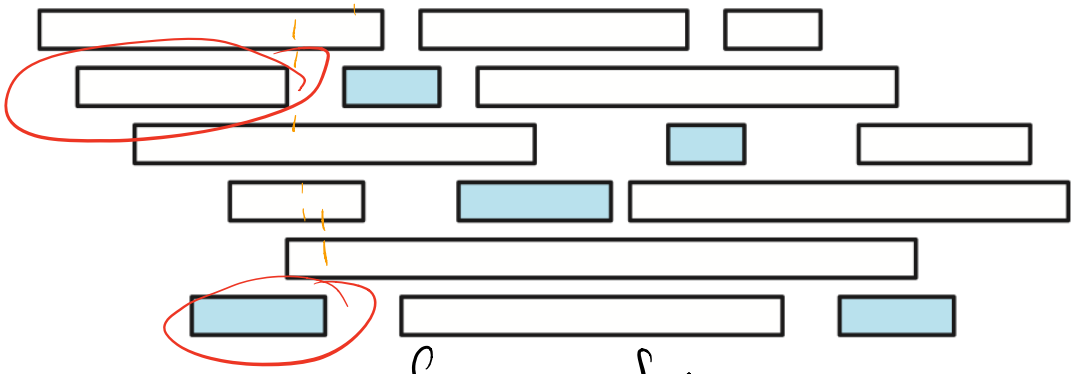
S[1..n]
F[1..n]

starting time of movie i
finishing time of movie i

$x_1, \ldots, x_n$

$F[x_i] \leq S[x_{i+1}]$

$F[i] \geq S[i]$ (always)



Let f_1 finishes first

$O_1, \ldots, O_k$

$S(O_2) \geq F(O_1)$

$S(O_2) \geq F(f_1)$

$O_2, \ldots, O_k$

f_2 finishes first, starts after f_1

Common pattern

→ at each step use "greedy heuristic" to select next element

→ exchange argument → optimal schedule solution ... greedy schedule solution

---

Companies A B C D

Students a b c d

internship interviews

A: a c b d

B: c a d b

C: a c d b

D: a d b c

a: C B D A

b: C D A B

c: D B A C

d: A B C D

Matching: each co ↔ one student

Unstable: Student x is matched w/ co X matched Y

if y prefers X to y
& prefers y to X
Y prefers x to y
X prefers y to x

unstable

if student X prefers Y to X
co Y prefers x to y

unstable

Stable assignment → not unstable

For any list of prefs
∃ a stable matching
and can be found in $O(n^2)$ time.

Each round

1. Find an unmatched company
2. Offer to next best person on their list not yet offered
3. Student accept if unmatched
   also accept if matched to worse co (reject old co)

reject if matched to better co.