Chapter 4. Greedy Alg.

failure case:

coin 1, 10, 25, 100

Target number T use min # coins to get T.

Greedy: largest coin.

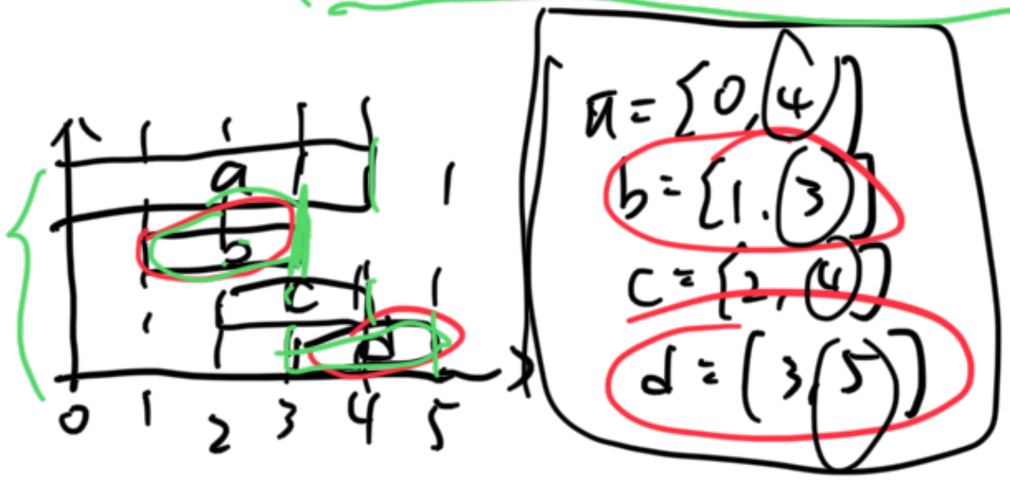
T=20 $10+10=20$ ok

T=30 $25+1+1+1+1+1=30$ 6 coins. GD.

$10+10+10=30$ 3 coins. optimal.

4.1 Scheduling:

Class 1, 2, ..., n each class starting time s(i)
ending time f(i).



Goal: schele max # classes that don't have overlap.

Idea: schedule class 1-by-1 according to some ordering.

① (failure case) Sort Starting time.

a → 1 class X

sorting by finishing time ✓

b, a, c, d

GD: Sort classes by finishing time $\pi_1, \pi_2, \dots \pi_n$: $f(\pi_1) \le f(\pi_2) \dots \le f(\pi_n)$

b a c d

A-finish = 0

$A \leftarrow \emptyset$

$O(n)$ { for $i = 1 \sim n$

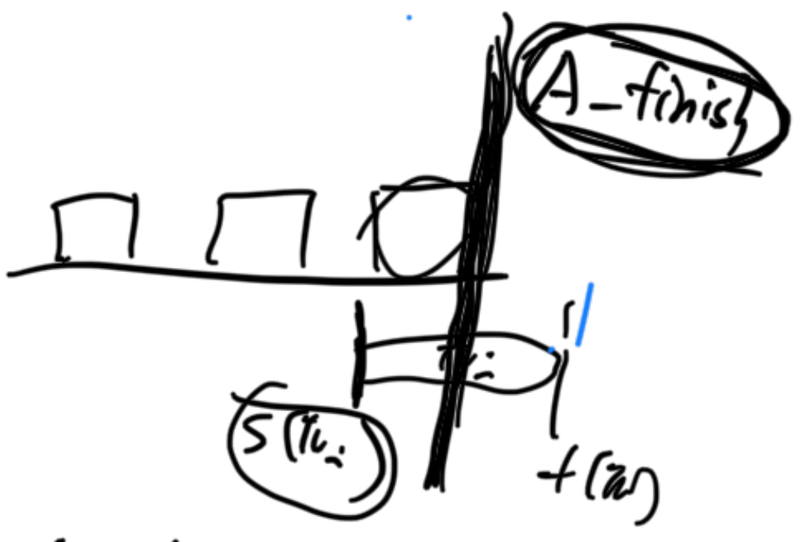
$O(1)$ { if π_i is not overlapping with A → if $s(\pi_i) \ge$ A-finish.

$A \leftarrow A + \{\pi_i\}$.

A-finish = $f(\pi_i)$

Time: $O(n \log n)$

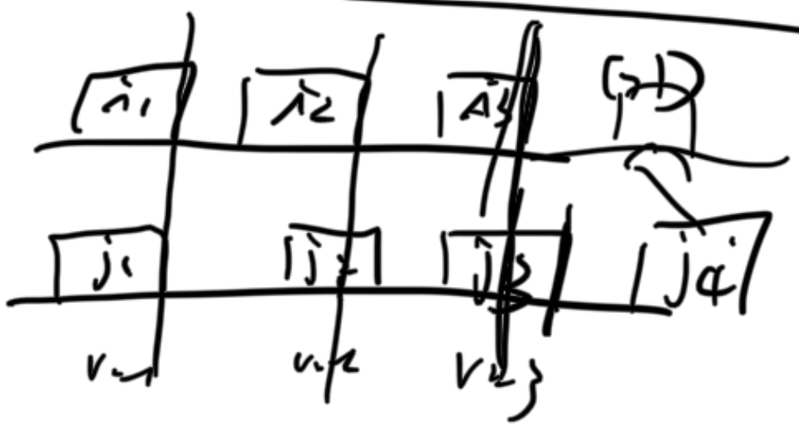
[sorting].

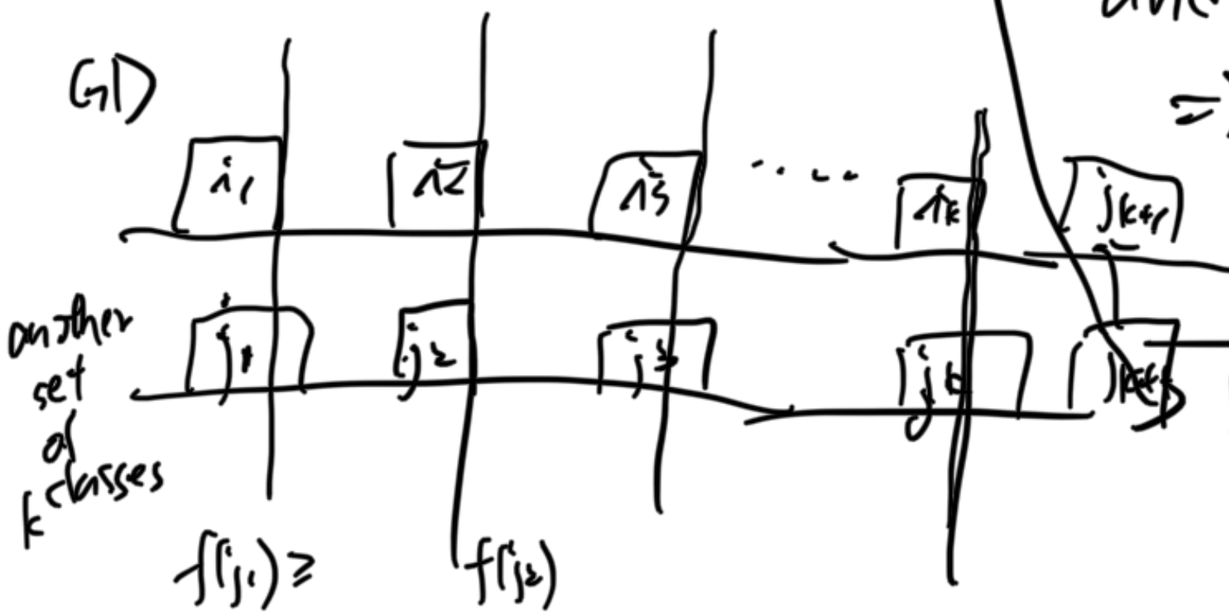


Thm: GD Alg outputs the optimal solution

pf: Let $\{i_1, i_2, \ldots, i_k\}$ is ...

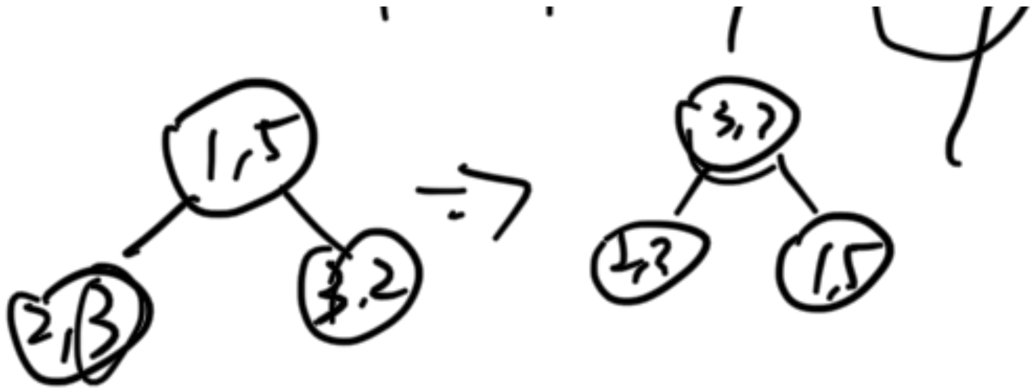
Claim: If $\{j_1, j_2 \ldots, j_k\}$ is another non-overlapping set.
then $f(i_r) \le f(j_r)$ $\forall r = 1, \ldots, k$.





(by contradiction)
$\exists \{j_1, \ldots j_k, j_{k+1}\}$ non-overlapping
$\Rightarrow f(i_k) \le f(j_k) \le s(j_{k+1})$
(claim)
and j_{k+1} will be checked after i_k in GD
$\Rightarrow$ GD will pick j_{k+1}
$\{i_1, \ldots i_k, j_{k+1}\}$.

pf by induction.
$k=1$ GD chooses i_1 with earliest finishing time. $j_1 \in$ another ok
Assume $f(i_r) \le f(j_r)$ $\forall r \le m$
when $m+1$, want to show $f(i_{m+1}) \le f(j_{m+1})$.
Based on induction assumption
$f(i_m) \le f(j_m) \Rightarrow$



class j_{m+1} is not overlapping
$\{i_1, \cdots, i_m\}$

since GD pick by finishing time ordering.

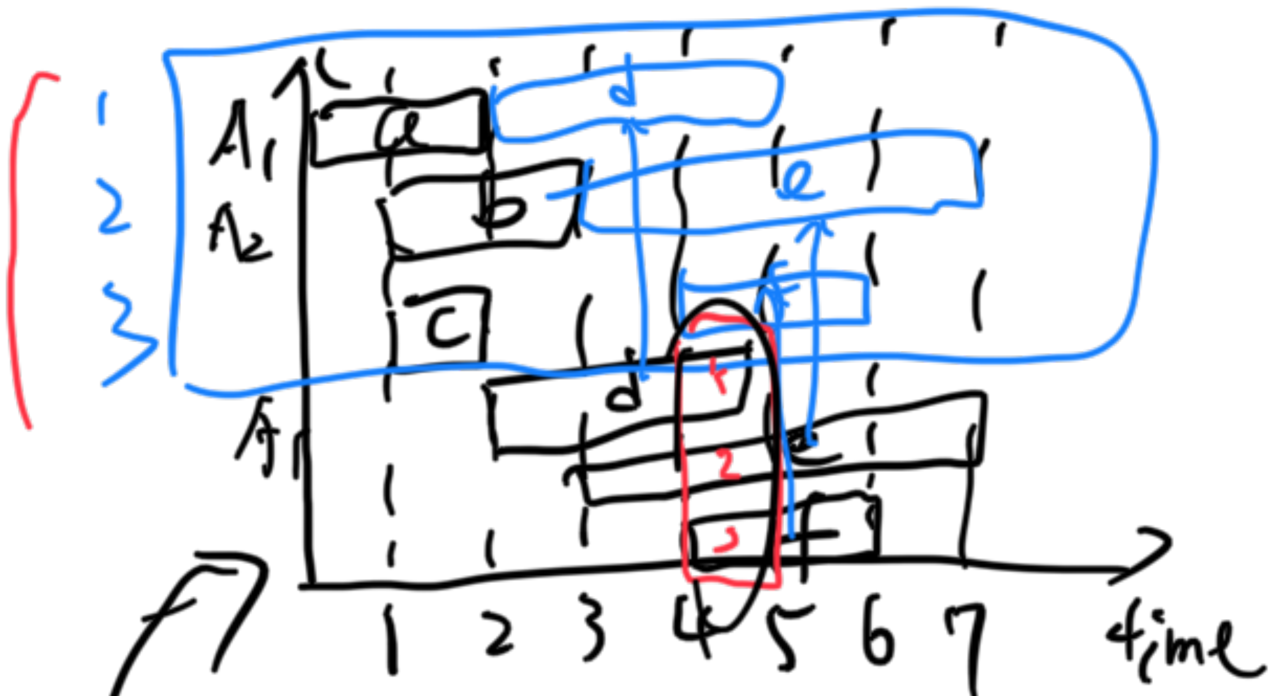
if $f(j_{m+1}) < f(i_{m+1})$

then GD will pick j_{m+1}. ✓

Interval partition:

class $1 \sim n$

Find min # classrooms to schedule all classes.



Idea: Sort classes in some ordering
schedule them 1-by-1.

Sort by finishing time.

Sorting by starting time.

Greedy Alg:

Sort classes by starting time $\pi_1, \pi_2, \cdots, \pi_n$ s.t. $s(\pi_1) \le s(\pi_2) \le \cdots \le s(\pi_n)$.

a, b, c, d, e, f

A_j.finish

$d = 0$ (d: # rooms).

for i: $1 \sim n$

If π_i is non-overlapping with a room k.

$A_k \leftarrow A_k + \{\pi_i\}$ (A_i.finish $= f(\pi_i)$)

else

create a new room $A_{d+1} = \{\pi_i\}$ → $O(\log n)$

A_d ...

$d = d+1$.

① store finishing time for each A_i_finish.

for $j \in 1 \sim d$
  if $s(x_i) \le A_j$-finish ····· $O(n) \Rightarrow O(n^2)$ for (>1).

② Use min heap to store $\{A_1\text{-finish}, A_2\text{-finish} \cdots A_d\text{-finish}\}$.

→ $O(1)$ to find $\min_j A_j$-finish.

check whether x_i fits in A_j.

→ Update a value in the heap: $= O(\log n)$

Correctness:

Overall: $O(n \log n)$.

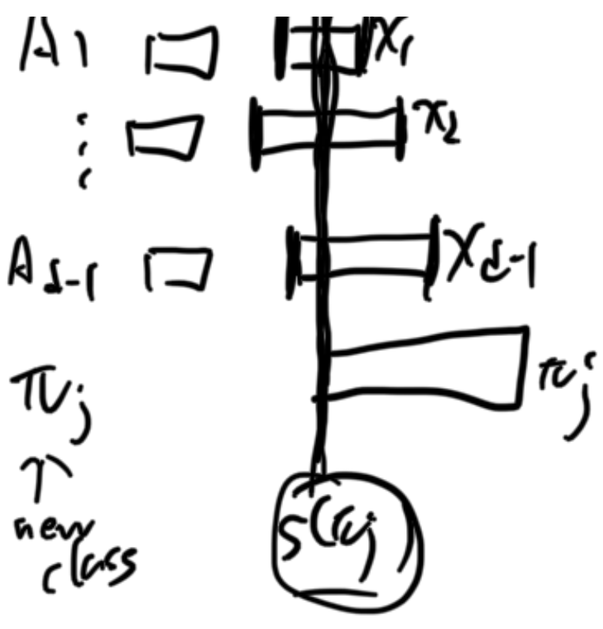
Define "depth": max # classes overlap at any time.

depth is a lower bound of # rooms we need.

Thm: Greedy Alg outputs d = depth.

pf: When GD create a new room d, (schedule x_j)

$s(x_j)$ ...



π_j overlaps with room $1, 2, \cdots, d-1$

$\Rightarrow$ class $x_1, x_2, \cdots, x_{d-1}$

Since we sort by starting time

$\forall i = 1, \cdots, d-1, \; s(x_i) \le s(\pi_j)$.

$\Rightarrow s(x_i) \le s(\pi_j) < f(x_i) \quad \forall i$

$\Rightarrow d$ classes overlap at $s(\pi_j)$

$\Rightarrow$ depth $\ge d$