Greedy Algorithms

Tuesday, April 7, 2026 9:21 AM

Announcements.

MT2 on April 13, Monday

Conflict " " 14, Tuesday

↳ (till 6pm by April 10).

GREEDY ALGORITHMS:

Solve optimization problem (min/max ...)

Iteratively by taking "local optimal steps".

Adv: Simple & Fast.

Disadv: Can be wrong! Needs "Proof of correctness".

Prob 1: Interval / Job scheduling:

Input: Given a set of n intervals

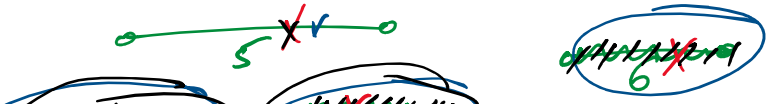
$\{[s_1, f_1], [s_2, f_2], \ldots, [s_n, f_n]\}$

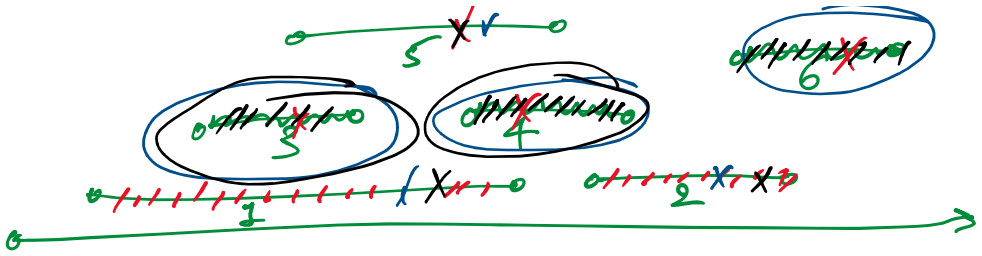
s_i: start time; f_i: finish-time.

Goal: Find maximum # of non-overlapping intervals.

eg.

OPT: 3





Greedy strategy (GS) 1: job w/ earliest start time.

2 X

GS 2: job w/ shortest length

In above example: 3 Yay! X

GS 3: job w/ minimum # overlaps.

2 ✓

In the first example 3 ✓ Yay yay!

Gives 3 X

GS 4: job w/ earliest finish time WORKS!

Intuition: then max time left for other job

High-level Greedy Alg: // o/p is in I.

① $R = \{[s_1, t_1], \ldots, [s_n, t_n]\}$; $I = \{\}$.

② while ($R \neq \emptyset$) {

③ Pick $[s_i, t_i]$ w/ earliest finish time. by some greedy strategy.

$R = R \setminus \{[s_i, t_i]\}$

③ Pick $[s_i, f_i]$ of some ...

④ $I = I \cup \{[s_i, f_i]\}$

⑤ $R = R \setminus \{[s_i, f_i]$ & all of it's overlapping intervals$\}$

}

Proof of Correctness: (By contradiction + exchange argument)

Assumption I: o/p of any greedy Al'g.

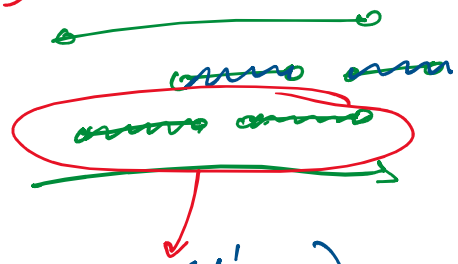
$\neq$

I^*: OPT sol'n.

Either I^* is unique.

OR There are multiple OPT sol'ns.

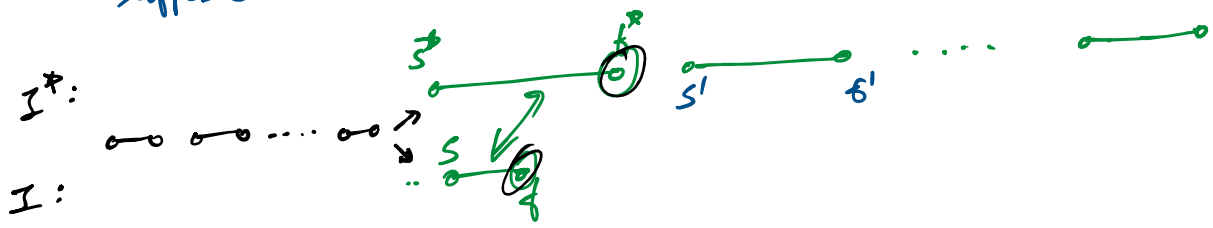
Then among these pick one w/ minimum total finish time $\left(\min: \sum_{[s',f'] \in I^*} f'\right)$



Let $[s, f]$ be the first interval in I

$[s^*, f^*]$ " " " I^*

Suppose: $[s, f] \neq [s^*, f^*] \Rightarrow f < f^*$ ($\because$ Greedy choice)



Exchange Arg:

$\forall [s', f'] \in I^*$, $[s', f'] \neq [s^*, f^*]$

We have, $f < f^* < s' \Rightarrow [s, f]$ & $[s', f']$ do not overlap.

$\Rightarrow I^* \setminus \{[s^*, f^*]\} \cup \{[s, f]\}$ is also OPT.

AND w/ smaller total finish time

AND w/ smaller total finish time

Contradiction! ∎

☆Alternate Proof of Correctness: (By Induction)

- ★ Base case: $n=1$, then greedy = opt.
- ★ Inductive Hypothesis: Suppose o/p of Greedy Al'g = OPT ∀ i/p w/ $n < k$
- ★ Induction: Let i/p be $T = \{[s_1,t_1], \ldots [s_k,t_k]\}$

Let I: o/p of Greedy Al'g. & $|I| < |I^*| \rightarrow (*)$

I^*: opt sol'n

Exchange:

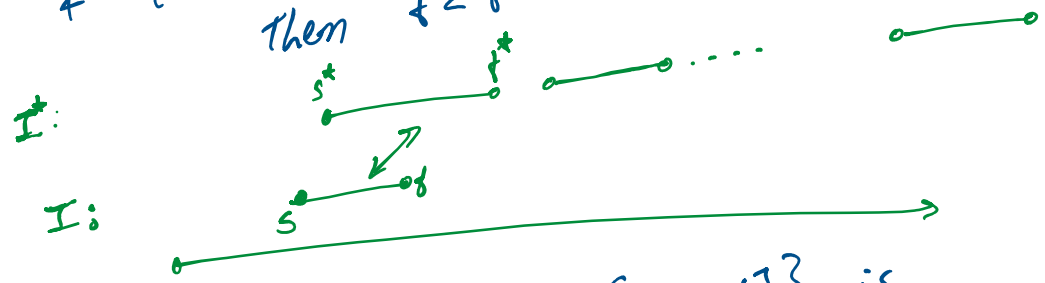
Let $[s,t]$: first interval in I

$[s^*,t^*]$: " " " I^*

$T' = T \setminus \{[s,t]$ & all intervals overlapping w/ $[s,t]\}$

& $[s,t] \neq [s^*,t^*]$

Then $t < t^*$



Then, $\tilde{I} = I^* \setminus \{[s^*,t^*]\} \cup \{[s,t]\}$ is another OPT.

& $I' = \tilde{I} \setminus \{[s,t]\} \subset T'$

→ Now, J = o/p of Greedy Al'g on T'

Then ① $|J| = |I| - 1$

② J is opt for T' by I.H. ∵ $|T'| < k$.

valid sol'n for T'.

Then ② J is opt for I'.

⇒ Note that, I' is a valid sol'n for I'.

$\therefore \quad |J| \geq |I'|$

Then $|I| = |J| + 1 \geq |I'| + 1 = |I^*|$ (∵ by construction of I')

$\Rightarrow |I| \geq |I^*|$, contradiction to ⑧ ∎

Scheduling: Minimize Wait Time

Tuesday, April 7, 2026 9:22 AM

Given: Processing time $p_1, p_2, \ldots, p_n$

Find a schedule (ordering) that minimizes (total wait time / completion time) cost.

eg. 3, 1, 5, 2

cost (3, 1, 5, 2) = (0) + (3) + (3+1) + (3+1+5) = 16

cost (3, 1, 2, 5) = (0) + (3) + (3+1) + (3+1+2) = 13

cost (1, 3, 2, 5) = (0) + (1) + (1+3) + (1+3+2) = 11

cost (1, 2, 3, 5) = (0) + (1) + (1+2) + (1+2+3) = 10

Observation: $p_1, p_2, \ldots, p_n$

min: $\text{cost}(p_1, \ldots, p_n) = (0) + (p_1) + (p_1+p_2) + \cdots + (p_1 + \cdots + p_{(n-1)})$

$= (n-1)p_1 + (n-2)p_2 + \cdots + p_{(n-1)}$

Greedy Alg: schedule in the increasing order of processing time.

Proof of correctness:

OPT: $p_1^* \; p_2^* \; \ldots \; p_i^*, p_{(i+1)}^* \; \ldots \; p_n^*$

(1)

OPT: $p_1^* \; p_2^* \; \ldots \; p_i^* \; p_{(i+1)}^*$

suppose: $p_i^* > p_{(i+1)}^*$ ①

opt-cost: $(n-1)p_1^* + \ldots + (n-i)\,p_i^* + (n-i-1)\,p_{(i+1)}^* + \ldots + p_{(n-1)}^*$

Exchange:

New: $p_1^* \; p_2^* \; \ldots \; p_{(i-1)}^* \; p_{(i+1)}^* \; p_i^* \; p_{(i+2)}^* \ldots p_n^*$

NewCost = $(n-1)p_1^* + \ldots + (n-i)\,p_{(i+1)}^* + (n-i-1)\,p_i^* + \ldots + p_{(n-1)}^*$

$$\begin{aligned} \text{OPTCost} - \text{NewCost} &= \left((n-i)\,p_i^* + (n-i-1)\,p_{(i+1)}^*\right) \\ &\quad - \left((n-i)\,p_{(i+1)}^* + (n-i-1)\,p_i^*\right) \\ &= p_i^* - p_{(i+1)}^* > 0 \end{aligned}$$

$\Rightarrow$ OPTCost > NewCost. A contradiction ∎