Greedy Algorithms

Change for Rs. R

smallest $\quad$ largest

$C_0, C_1, \ldots \quad C_k$

$G \to \{ n_0, \ldots, n_k \}$ $\quad n_k = \left\lfloor \frac{R}{C_k} \right\rfloor$

$G = \{ n_k, n_{k-1}, \ldots, n_0 \}$ $\quad$ Greedy Algo is defined

$C_k, \ldots \quad C_0$

Prove that this is optimal

Assume an optimal sol^n

$O = \{ m_k, m_{k-1}, \ldots m_0 \}$

Assume that O differs from C at ith posn

greedy n_i coins
optimal m_i coins

} denomination c_i

$c_k, \ldots, c_{i+1} \rightarrow c_i$

$n_k = m_k$ ... $n_{i+1} = m_{i+1}$ ... $\underline{n_i}$ m_i

$\boxed{n_i > m_i}$ greedy property

Create O' st $\boxed{m_i = n_i}$ $\rightarrow$ adjust coin from the later denominations -->

$n_i > m_i$ diff. is atleast 1

O m_i

$\downarrow$

O' n_i

+1 coin of denomination c_i → I need atleast 2 coins of lower denⁿ

You're to remove some coins from older denⁿ

c_{i+1} $\{c_{i-1}, \ldots c_0\}$

$c_1 \geq 2c_0$

$c_2 \geq 2c_1$ . . .

$c_i \geq 2c_{i-1}$

① Finding counter-examples (to prove that this is not optimal)

② Prove (to show that this is optimal)

Interval Scheduling with deadline Knapsack

Lab Problem :→ System $e(k,p)$

Reaction N

$k = 1, \ldots, N$

Prob. of success of system

assignment of units of catalyst

$p = 1, \ldots, G$

N

$e(k,p)$ is non decreasing for p (given k)

Greedy Approach :-> The reaction with $e(k,p)/e(k,p-1)$ is max for any p

3

5

0.1	0.3	0.5	0.6	0.6
0.1	0.2	0.4	0.6	0.8
0.2	0.4	0.4	0.7	0.9

0.3 R1 1, 1

0.2 R2 1, 1

0.2 R3 1

12×10^{-3}

Now take the reacn with max rise in prob

Counter example

	3					
	6					
→ 1	$\boxed{0.5}$	0.5	0.6	0.8	0.9	0.9
→ 4	0.4	0.4	0.6	$\boxed{0.7}$	0.8	1.0
1	$\boxed{0.5}$	0.5	0.7	0.8	0.9	0.9

$0.7 \times 0.5 \times 0.5$

$= \boxed{0.175}$

0.175

R1 1, 1, 1, 1 0.8

R2 1 $\times 0.4$

R3 1 $\times 0.5$

$= 0.16$

doesn't give an optimal sol^n.

Choosing a counter-example

Try to find an example where some possible arbitrary tie breaking doesn't lead to an optimal soln.

Or

In your greedy - specify tie-breaking

Problem 6 ⇒ ∃ a train station, train are arriving at various time $a_1 \ldots a_n$ @ strictly increasing order

Whenever a train arrives, one staff is required to oversee.

a staff can atmost stay for αn hours.
[9:00 AM - 10:AM]

Find ~~min~~ an algo to do an assignment with min. # of staffs required, and the time-period of two staff doesn't overlap.

$\{ a_1 \quad \ldots \quad a_n \}$

8:30 9:00 9:15 9:30 9:55 10:30

find a greedy algo to do the scheduling of staff — atleast n staff

Algo ↳ 1. The first staff comes for $[a_1, a_1 + 1]$

The next staff comes at the next a_i after $a_1 + 1$

[8:30 9:30] [9:55 10:55] 2 staff

Algo 2 find an interval (how long) where the max. number of trucks arrive. Assign a staff to that. Now recursively do that for the remaining arrival times.

Which one is incorrect?

Algo 2.

Counter example

9

size 4

9:30 10:15

2 arrivals

2

9:30 10:15 10:30 10:45 11:00 11:30 11:45

10.20

4

4

4

2 staffs.

assign a staff

Example

10:45 11:00 11:30 11:45

4 arrivals

assign a staff

algo 2 is not optimal

4

9:30 10:15 10:20 10:45 11:00 11:30 11:45

2 4 1

6

10:00 10:15 10:30 10:45 10:50 11:05 11:10 12:00

3 5 1 2

10:00 1 11:00 2 12:00

3 staffs Greedy

Optimal

Prove that Algo 1 is optimal.

$\{a_1, \ldots a_n\}$

Greedy algo

starts from the first arrival time

& assigns a staff to $[a_1, a_1+1]$

next arrival - -

Suppose it creates k intervals

$G = \{g_1, \ldots g_k\}$

min. the staff

Assume there is an optimal soln

$*$ Covers all the arrival times

$O = \{o_1, \ldots o_l\}$

$\{l < k\}$ $\{l \leq k\}$

$*$ No two interval(s) schedule are overlapping

$G = \{ g_1 \ldots g_k \}$

$O = \{ o_1, \ldots o_l \}$ $l \le k$

[Among all the optimal sol^n s, suppose we take the one which is identical to G for the longest seq. starting from the beginning.]

$\Rightarrow$ There cannot be an optimal sol^n that aligns with G for one more interval.

$O_{100} = \{ o_1, \ldots o_l \}$

matches G for first t time-points.

Try making O similar to G at $t+1$ th time point without losing optimality

$\Longrightarrow$ Contradiction

$G \quad \{g_1, \ldots g_t, g_{t+1}, \ldots g_k\}$

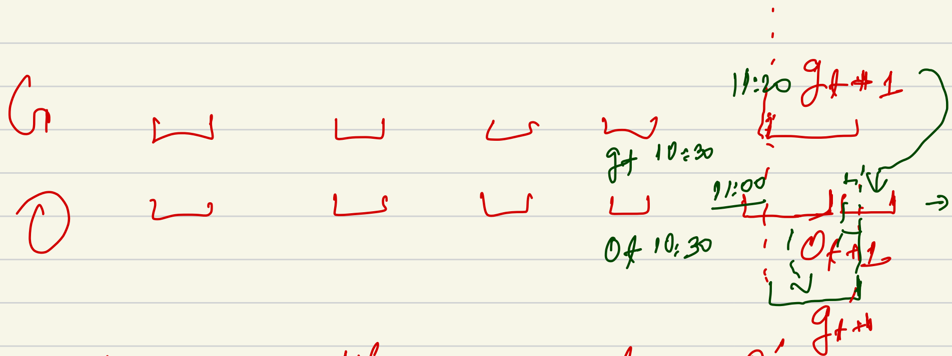
$O = \{o_1, \ldots o_t, o_{t+1}, \ldots o_k\}$ covers all trains, no overlap

Suppose G & O are ordered wrt their left endpoint

$o_{t+1} \neq g_{t+1}$

g_{t+1} $\quad o_{t+1} > g_{t+1}$ X

o_{t+1} $\quad g_{t+1} > o_{t+1}$



if there is an overlap

Now modify O to O' s.t.

$$\boxed{O_{t+1} = g_{t+1}}$$

No change anywhere else

still get optimal soln

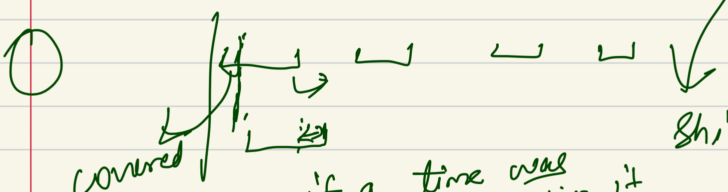
⇓

contradiction

[Hence, the proof!]

* Covers all the points ✓
* No overlap ✓

Can I ensure optimality?



if a time was covered earlier, it is still covered

Shift O_{t+2} to the right until no overlap

⋮

Keep doing this for all sub. intervals

Problem :=> Convocation time

n people $\quad h_1, \ldots, h_n$

gown size $\quad s_1, \ldots, s_n$

Let's say ith person gets $\alpha(i)$-th gown

$$\text{min.} \quad \frac{1}{n} \sum_{i=1}^{n} |h_i - s_{\alpha(i)}|$$

1 2 3 4 5

$\updownarrow$ $\updownarrow$ $\updownarrow$ $\updownarrow$ $\updownarrow$

5 6 7 8 9

4

an assignment of gowns to students to optimize this quantity

Greedy Algo

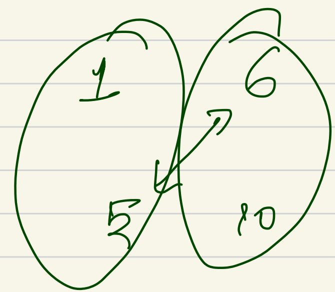
Algo 1. Sort the heights & gown sizes

assign in the order [shortest person gets the shortest gown, ...]

Algo 2: Find the person & gown with min. heigh difference. Assign the gown to this person. Repeat until everyone has a gown.

$$\min_{\forall i} |h_i - S_{\alpha(i)}|$$

Not optimal?



$$\begin{array}{cc} 1 & 6 \\ 5 & 10 \\ \hline 4 & 4 \end{array} \rightarrow 4$$

Algo 2.

counter example

$$\begin{array}{cc} 1 & 6 \\ 10 & 5 \\ \hline \end{array}$$

$$9 + 1 = \frac{10}{2} = 5$$

Prove that Algo 1 is optimal

Let's assume that people & gowns are sorted

$h_1, \ldots h_n \leftarrow$ height

$s_1, \ldots s_n \leftarrow$ sizes

Greedy

Optimal

for some inputs, greedy & Optimal may be same.

If Greedy sol^n is not optimal, there is atleast some input on which the greedy sol^n will differ from optimal sol^n.

Let's assume this is that input

Greedy : $(h_1, s_1), (h_2, s_2), \ldots (h_n, s_n)$

Optimal : $(h_1, s_{\alpha(1)}), (h_2, s_{\alpha(2)}), \ldots$

$(h_n, s_{\alpha(n)})$

Greedy (h_i, s_i)

· OPTIMAL (h_i, s_j) (h_k, s_i)

$\mathrm{cost}(O')$

$= \mathrm{cost}(O)$

$+ |h_i - s_j| + |h_k - s_i|$

$- |h_i - s_i| - |h_k - s_j|$

$h_1, \ldots, h_{i-1}$ same

$s_1, \ldots, s_{i-1}$ differ

ith

h_i

Prove that this is ≥ 0 always

s_i s_j

Where would s_i go in O?

some $h_k \geq h_i$

$s_i \leq s_j$ $h_i \leq h_k$

$s_i \leq s_j \leq h_i \leq h_k$ $h_i \leq h_k \leq s_i \leq s_j$

Prove that this doesn't increase the cost

- Modify O to do this assignments

O' (h_i, s_i) (h_k, s_j)