29.11.2023.

NOTIȚE CURS 9 (amf.)

METODA GREEDY - p2

2) Programarea spectacolelor într-o sală $[s_i, f_i) \Rightarrow f_i - s_i$

1 sală <— n spectacole

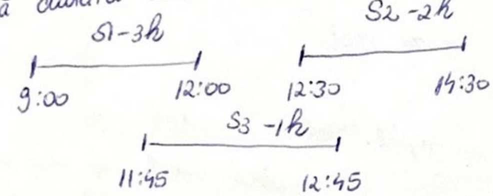
$[s_1, f_1), [s_2, f_2) \dots [s_n, f_n)$

[9:30, 10:30) [10:30, 12:00)

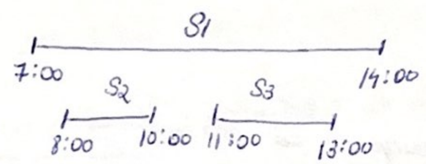
? Nr. max de spectacole fără suprapunere

Criterii posibile

a) după durată cresc. => S_3 (optim S_1, S_2)



b) după ora de început cresc. => S_1 (optim S_2, S_3)



c) după ora de terminare / final cresc. $f_i - s_i + s_i$

pt. ex de la a: S_1, S_2 (corect)

pt. ex de la b: S_2, S_3 (corect)

Alg. Greedy:

1) Sortăm spectacolele cresc. după ora de final

2) Planificăm primul spectacol

3) Pt. restul spectacolelor verif. dacă sp. curent nu se suprapune cu ult. sp. planificat și în caz afirmativ îl planificăm

$$O(\underset{①}{n\log_2 n} + \underset{②}{1} + \underset{③}{n}) = O(n\log_2 n)$$

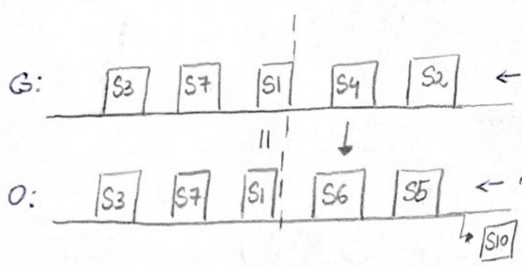
Demonstrația corectitudinii (folosim exchange argument)

Ex: $n = 7$ spectacole

S1: 10:00 - 11:20
S2: 9:30 - 12:10
S3: 8:20 - 9:50
S4: 11:30 - 14:00
S5: 12:10 - 13:10
S6: 14:00 - 16:00
S7: 15:00 - 15:30

1) → S3, S1, S2, S5, S4, S7, S6 (sortate după timpul final)
2) → S3
3) → S3, S1, S5, S7
↳ am putea să punem și S6

G: (sol. Greedy)
pp. că nu este optimă => ∃ O: (o soluție optimă, O ≠ G)



← m spct.
← n spct. putea fi cazul S6 S5
S4 → S6
S10 - dacă ar ∃ => ar fi fost și în G pt. că $s_{10} <= f_5$ => ar fi inclus și S10
deci acest caz nu se poate întâmpla niciodată

$S_4 > f_1$ (corect)
! S4 se termină sigur înaintea lui S6 (sau eventual $f_4 = f_6$) pt. că altfel G l-ar fi ales pe S6 nu pe S4

~~1) m > n~~ niciodată pt. că n se presupune că este optimă
~~2) m < n~~ explicația cu S10
$\boxed{3)\ m = n}$ este corectă și dacă $G = O$ ($m = n$) => G este soluția optimă

3) Planificarea unor spectacole folosind un număr minim de săli. (o pb. mai avansată derivată din pb. 2) ↓după ultimul din sală

	C1	C2	C3
S1: [10:00 - 11:20)	Sala 1	Sala 3	Sala 1
S2: [9:30 - 12:10)	Sala 2	Sala 5	Sala 2
S3: [8:20 - 9:50)	Sala 1	Sala 4	Sala 1
S4: [11:30 - 14:00)	Sala 4	Sala 3	Sala 1
S5: [12:10 - 13:10)	Sala 1	Sala 2	Sala 2
S6: [11:15 - 13:15)	Sala 3	Sala 4	Sala 3
S7: [15:00 - 15:30)	Sala 1	Sala 1	Sala 1
	4 săli	5 săli	3 săli

Criterii

1) După ora de terminare cresc:

Sala 1: S_3 S_1 S_5 S_7
Sala 2: S_2
Sala 3: S_6
Sala 4: S_4

4 săli

2) După durată cresc.

mai rău (5 săli)

3) După ora de început cresc.

mai bun (3 săli)

Algoritmul Greedy

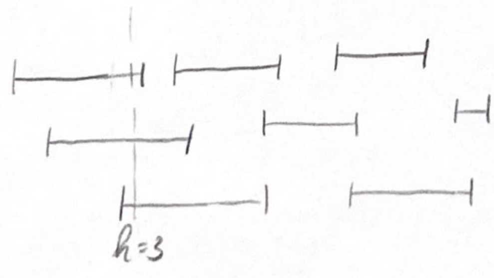
1) Sortăm cresc. spectacolele după ora de început
2) Programăm primul spectacol în sala 1
3) Pt. fiecare spectacol rămas căutăm prima sală în care ar putea a fi programat sau dacă nu ∃ nicio astfel de sală îi alocăm o sală nouă.

! coadă cu priorități - structură de date

min priority → extrage în O(1) tuplul cu cheia minimă

Demonstrația corectitudinii

1) Adâncimea unui șir de intervale de desfășurare a unor spectacole este dată / este egală cu nr. max. de spct. care se suprapun la un moment dat



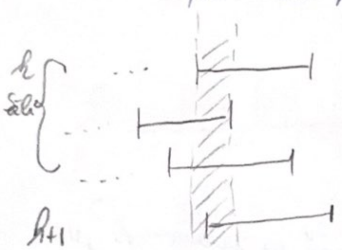
2| Orice planificare corectă a spct. folosește un nr. de săli ≥ adâncimea șirului de intervale

3| Alg. Greedy ~~nu programează~~ furnizează întotdeauna o programare corectă ⇒ alg. greedy va folosi cel puțin h săli

4| Pt. un șir de intervale cu adâncimea h, alg. Greedy va planifica spectacolele în cel mult h săli

3+4 ⇒ alg. optim

4: PP. prin absurd că alg. Greedy nu ar folosi cel mult h săli
Negare ⇒ pt. un șir de intervale cu adâncimea h, alg. Greedy ar folosi cel puțin (h+1) săli



↳ adâncimea = h+1 ⇒ contradicție cu