5/03/2024 | INIZIO MOD 2

FUTURI ARGOMENTI → TECNICHE ('PARADIGMI') X PROJECT ALGO.

- ZOOM SU TECNICHE ALGORITMICHE
  - ALGORITMI GREEDY
  - PROGRAMMAZIONE DINAMICA
  - ALGO. PER MAX FLOW
  - PROBLEMI N.P.-COMPLETI

ESAMI

- 6 APPELLI
- POSSO DECIDERE CHE MOD. FARE

ALGORITMI GREEDY → "GOLOSO": PRENDE TUTTE LE STIME.

VEDREMO 2 ALGORITMI GREEDY:

1° INTERVAL SCHEDULING → PROBLEMA DI "OTTIMIZZAZIONE"

- DATI DEL PROBLEMA:

  INSIEME DI "JOBS" (PROCESSI) CON: · $\hat{\imath}$ → ISTANTE DI INIZIO JOB
  · f → ISTANTE DI END JOB

- OBB.: IN UN INTERVALLO t E n JOBS, METTERE QUANTI PIÙ JOBS ϕ COMPATIBLE POSSIBLE

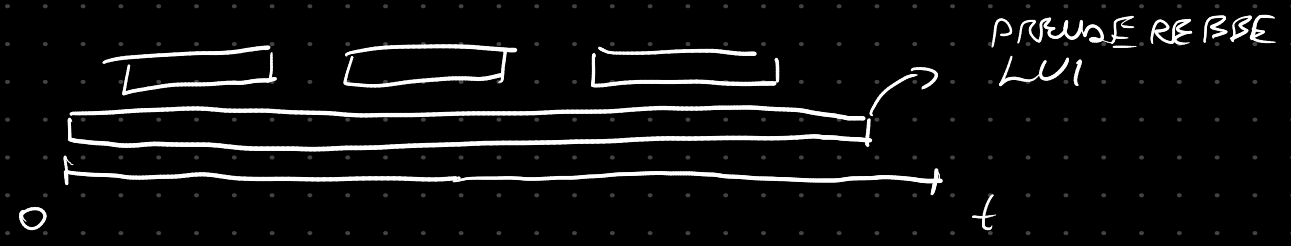
N.B. JOBS COMPATIBLE = JOBS CON INTERVALLI DI t IN COMUNE

ALGORITMO

- SORT JOBS IN UN ORDINE PRECISO,
  - ↳ QUALE?
- APPROCCIO GREEDY
  - ↳ CONTROLLO EACH JOBS IF ϕ COMPATIBLE

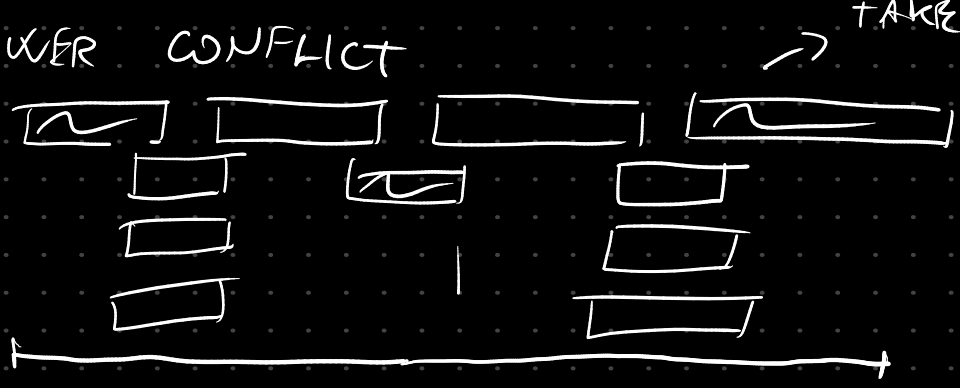
ORDINE SORT → ≠ OPPORTUNITÀ, MA 1 SOLA WORKS.

① EARLIEST START TIME NOP



② SHORTEST INTERVAL

③ FEWER CONFLICT



IL MIGLIORE È SORT BY FINISH-TIME

QUINDI...

ALGORITMO

S = START TIME

E OTTIMIZZARE

EARLIEST-FINISH-TIME-FIRST $(n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n)$

SORT jobs by finish times and renumber so that $f_1 \le f_2 \le \ldots \le f_n$.

$S \leftarrow \emptyset$. ← set of jobs selected

FOR $j = 1$ TO n

  IF (job j is compatible with S)

    $S \leftarrow S \cup \{ j \}$.

RETURN S.

× COSTRUTTO, BASTA RICORDARE $f_{i_{r-1}} = $ LAST JOB $\in S$

JOB $*$

IF $s_* \ge f_{i_{r-1}}$ -> COMPATIBLE

COSTO ? → CICLO DI CONFRONTI $= O(n)$

→ SORT $= O(n \log n)$

$O(n \cdot \log n)$

× CHÈ È IL MIGLIORE ?

DIM

SIA

- $i_1, i_2, \ldots, i_k$ SET JOB CHOOSE DA GREEDY
- $J_1, J_2, \ldots, J_m$ " " " OPTIMAL (ALG. OTTIMO)

LEMMA: $\forall r = 0, 1, \ldots, k \quad f(i_r) \le f(J_r)$

INDUZIONE

$r = 1$ -> OVVIO.

$r > 1$ → IL TEMPO $f(i_r)$ DEVE $\le f(J_r)$

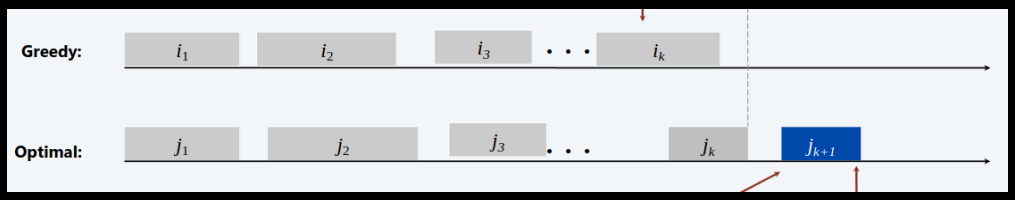
ALTRIMENTI → GREEDY COULD TAKE

ORA, X ASSURDO:

- GREEDY NOT OTTIMALE

⇓

QUINDI $m > k$, MA:



↳ SE CI FOSSE IN T_{k+1}, QUESTO VERREBBE PRESO DA GREEDY

QUINDI:

GREEDY ALGO. OTTIMALE — CONTRADDIZIONE

2° INTERVAL PARTITIONING

- INPUT → SET JOBS CON (S,F) VALORI DI PRIMA

- SOL. "FLESSIBILE" → SUBSET OF JOB ∈ IN $C = 1, 2, \dots, m$ ∀ CONTIENE PART OF JOB.

- OUTPUT → MIN m. CLASS (C), OGNUNO CONTIENE JOB. COMPATIBLE.

ALGORITMO → SIMILE A SCHEDUL.

- SORT JOBS → ORDINE DA TROVARE
- CONFRONTO TRA JOB - CLASS
  ↳ IF ∄ COMPAT.:
  • JOB ∪ CLASS

  IF NOT:
  • CREA NUOVA CLASS → CLASS' ∪ JOBS

ORDINE SORT → SORT BY STARTING-TIME

ALGORITMO

EARLIEST-START-TIME-FIRST $(n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n)$

SORT lectures by start times and renumber so that $s_1 \leq s_2 \leq \ldots \leq s_n$.

$d \leftarrow 0$. ⟵ number of allocated classrooms

FOR $j = 1$ TO n

  IF (lecture j is compatible with some classroom)

    Schedule lecture j in any such classroom k.

  ELSE

    Allocate a new classroom $d + 1$.

    Schedule lecture j in classroom $d + 1$.

    $d \leftarrow d + 1$.

RETURN schedule.

- X OPTIMIZE CLASS = CODA PRIORITA' (KEY = F LAST JOB.)
- NEW CLASS = INSERT
- INSERT JOB K IN CLASS C = INCREASE KEY (K, C)
- COMPATIBLE: COMPARE S_K WITH FIND-MIN.

COSTO

- SORT → $O(n \log n)$
- OP U C. PRIORITA' → OP. TOTALI IS $O(n)$., OGNUNO IN $O(\log n)$

$TOT = O(n \cdot \log n)$

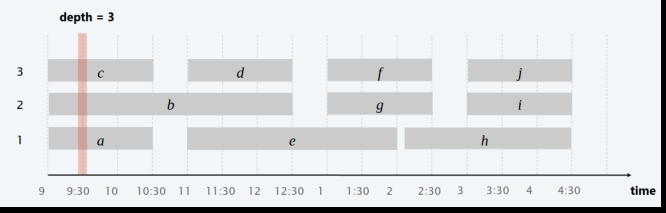
X CHÈ OPTIMAL?

DIM

Def. The depth of a set of open intervals is the maximum number of intervals that contain any given point.

Key observation. Number of classrooms needed ≥ depth.

Q. Does minimum number of classrooms needed always equal depth?
A. Yes! Moreover, earliest-start-time-first algorithm finds a schedule whose number of classrooms equals the depth.



⇒

Observation. The earliest-start-time first algorithm never schedules two incompatible lectures in the same classroom.

Theorem. Earliest-start-time-first algorithm is optimal.

Pf.

- Let d = number of classrooms that the algorithm allocates.
- Classroom d is opened because we needed to schedule a lecture, say j, that is incompatible with a lecture in each of $d - 1$ other classrooms.
- Thus, these d lectures each end after s_j.
- Since we sorted by start time, each of these incompatible lectures start no later than s_j.
- Thus, we have d lectures overlapping at time $s_j + \varepsilon$.
- Key observation ⇒ all schedules use ≥ d classrooms. ▪