

ESERCIZIO GREEDY 4

N compiti da svolgere i Compito i (a_i, b_i) con $a_i \leq b_i$ a_i, b_i interi

- Vogliamo eseguire tutti i compiti
- Ogni compito richiede 1 ufficio per essere svolto
- Non si possono svolgere 2 compiti che si sovrappongono nello stesso ufficio

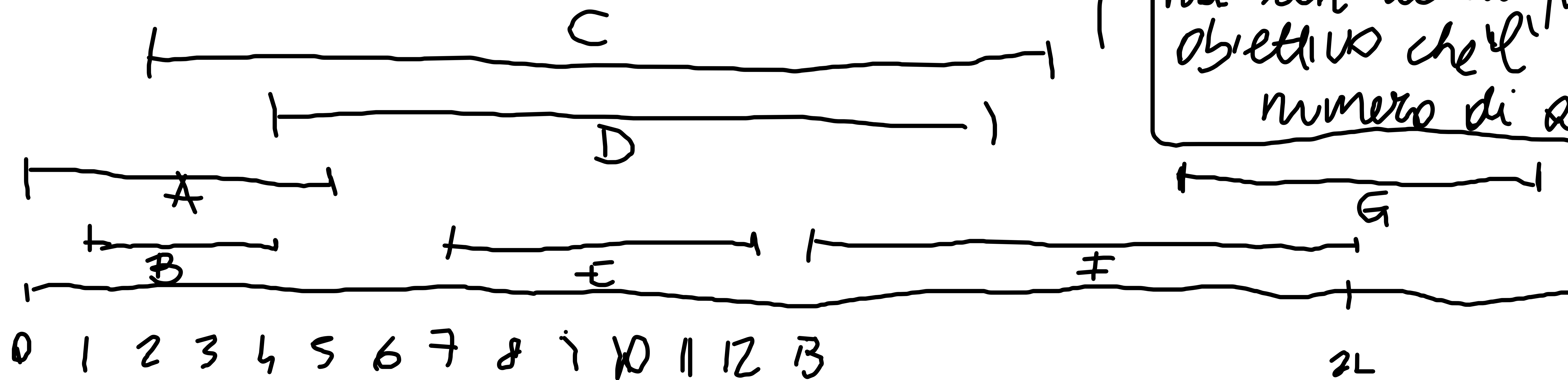
FUNZIONE
OBETTIVO: Minimizzare il numero di uffici necessari per svolgere tutti i compiti

	a	b
A	0	5
B	1	3
C	2	17
D	4	16
E	7	12
F	13	22
G	19	25

Idea 1: Applichiamo Selezione di attività ripetute

- ↳ 1) Ordinare le attività per tempo di completamento non decrescente
- 2) Scegliere attività t.c. $b_i \leq a_j$ per $j > i$
- 3) Ripetere aggiungendo un ufficio nuovo ogni volta finché non finiamo la lista di attività

	a	b	U
B	1	3	U ₁
A	0	5	U ₂
E	7	12	U ₁
D	4	16	U ₃
C	2	17	U ₄
F	13	22	U ₁
G	19	25	U ₂



NO OTTIMO!
 GREEDACTIVITY SELECTOR
 ha un'altra funz.
 obiettivo che è MAX
 numero di attività

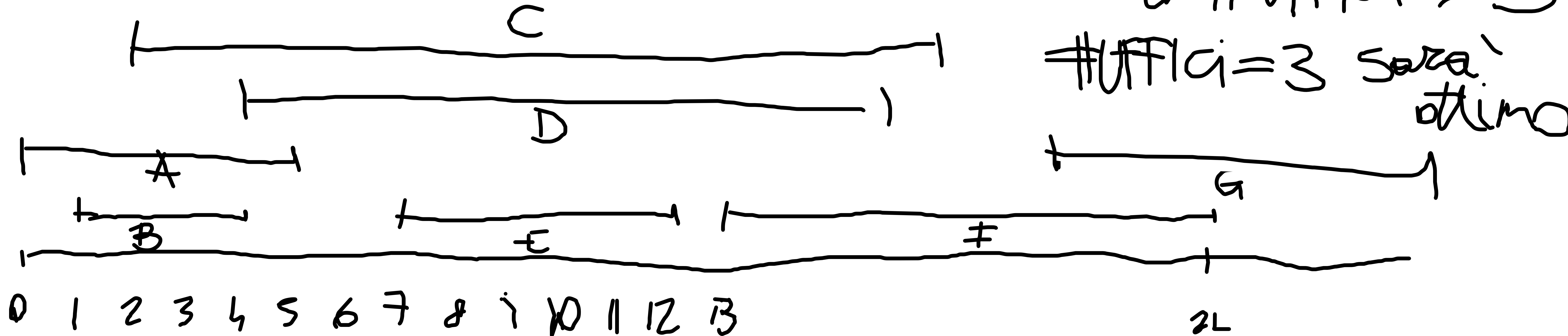
	a	b
A	0	5
B	1	3
C	2	17
D	4	16
E	7	12
F	13	22
G	14	25

Perché la soluzione non è ottima?

Il numero di attività mutualmente incompatibili determina un vincolo strutturale del problema

In questo caso abbiamo 3 attività a coppie incompatibili \Rightarrow il numero necessario di uffici per svolgere tutte le attività è 2

\rightarrow Il vincolo strutturale ci dice che ogni soluzione ammissibile avrà una proprietà $\# \text{Uffici} \geq 3$

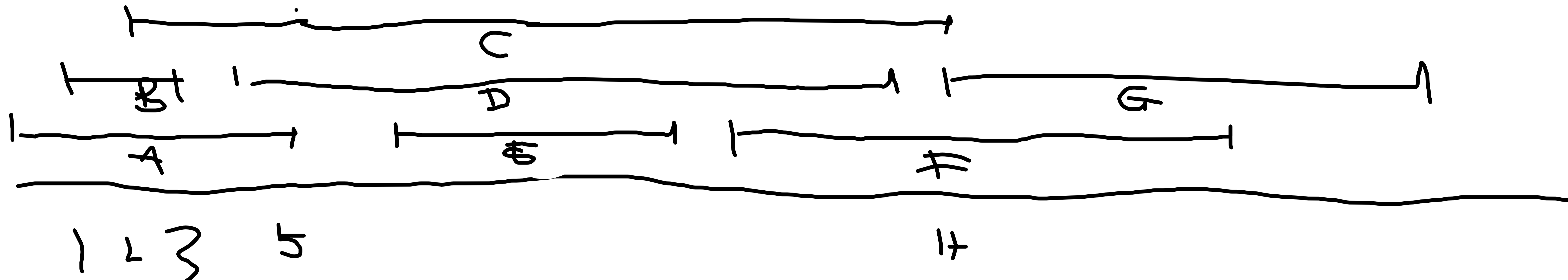


	a	b	U
B	1	3	U ₁
F	0	5	U ₂
A	7	12	U ₁
C	4	16	U ₃
D	2	17	U ₄
E	7	22	U ₁
G	14	25	U ₂

Richiesta di algoritmo.

- Ordina le attività per tempo di inizio non decrescente
- Finché l'insieme di attività non è vuoto
 - Estrai dalla lista l'attività che inizia per prima
 - Se un ufficio la può eseguire
Assegna l'attività a quell'ufficio
- Altrimenti aggiungi un nuovo ufficio ed assegna l'attività

	a	b
A	0	5
B	1	3
C	2	17
D	4	16
E	7	12
F	13	22
G	14	25



#UFFICI* = 3

Introdurre la k-esima risorsa nel momento in cui deve essere eseguita la k-esima attività fa sì che la procedura rispetti il vincolo strutturale \rightarrow che sia corretta

TIPS PER GREEDY

VINCOLO STRUTTURALE: Trova un vincolo strutturale t.c. ogni possibile soluzione debba avere un certo valore. Dimostra che l'alg. greedy porta sempre a quel valore.

SCAMBIO: Trasforma una soluzione ottima in una fornita dall' algoritmo greedy senza peggiorare la qualità.

GREEDY E' MEGLIO: Dopo ogni step dell' algoritmo greedy dimostra che la sol. greedy non e' peggiore di quella fornita da un qualunque altro algoritmo.

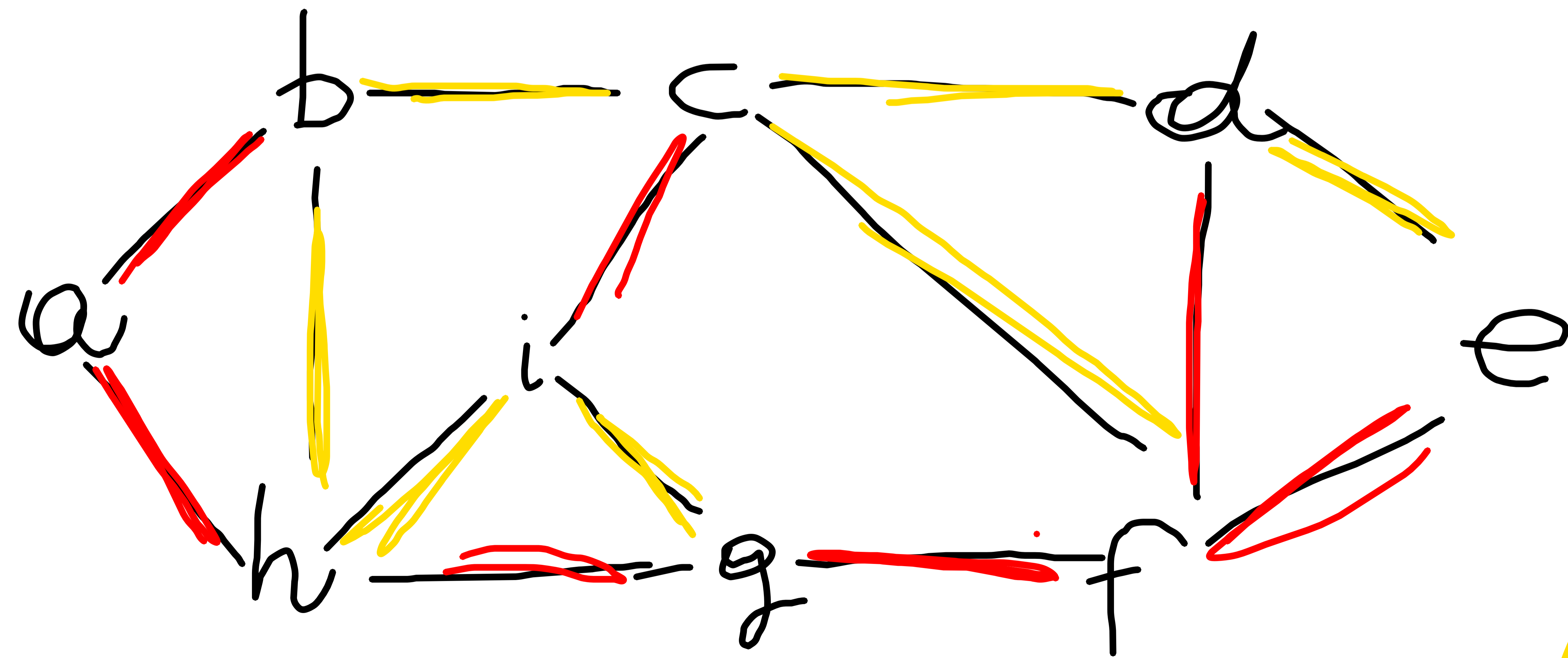
SOTTOSTRUTTURE

PREFISSI	$X[1, i]$
SUFFISSI	$X[i, n]$
SUBSTRING	$X[i, j]$

Esercizio: Un grafo G ha gli archi colorati di giallo e rosso. Trovare un algoritmo che restituisca un MST per G con il minimo numero di archi rossi nei casi in cui ?

1) G non è pesato

2) G è pesato



↳ Somma
al peso degli
archi rossi il max
dei pesi degli archi
(gialli)

→ Assegniamo peso
1 agli archi gialli
e 2 agli archi rossi
e usiamo Kruskal
o Prim.