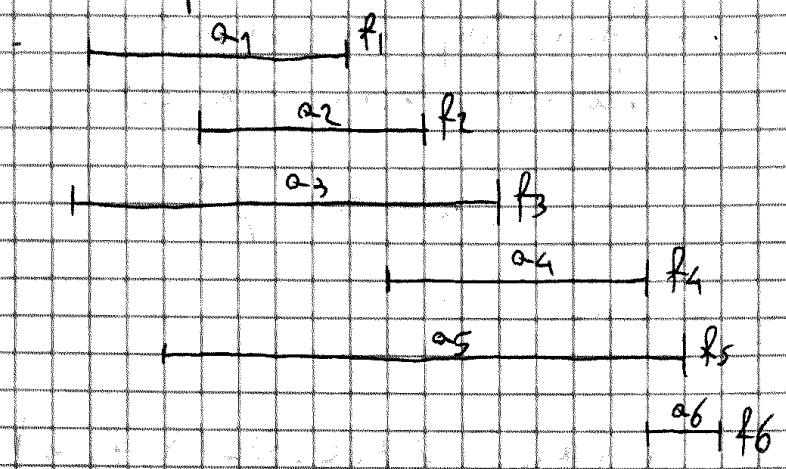
Esempio:



$A \leftarrow \{a_1\}$, $last \leftarrow 1$ — inizio

$A \leftarrow \{a_1\}$, $last \leftarrow 1$ — $m=2$

$A \leftarrow \{a_1\}$, $last \leftarrow 1$ — $m=3$

$A \leftarrow \{a_1, a_4\}$, $last \leftarrow 4$ — $m=4$

$A \leftarrow \{a_1, a_4\}$, $last \leftarrow 4$ — $m=5$

$A \leftarrow \{a_1, a_4, a_6\}$
$last \leftarrow 6$
$m=6$

Esempio: (domanda d'esame)

6 attività $s = (1, 5, 2, 3, 7, 10)$ $f = (3, 7, 9, 10, 11, 12)$

Determinare l'output di GREEDY_SEL, descrivendo brevemente l'algoritmo

$A \leftarrow \{a_1\}$, $last \leftarrow 1$

$A \leftarrow \{a_1, a_2\}$, $last \leftarrow 2$

$A \leftarrow \{a_1, a_2\}$

$A \leftarrow \{a_1, a_2\}$

$A \leftarrow \{a_1, a_2, a_3\}$, $last \leftarrow 5$

$A \leftarrow \{a_1, a_2, a_5\}$

Paradigma generale

Si basa su 2 proprietà

- Proprietà di scelta greedy

  la soluzione ottima può essere costruita componendo scelte "ardite"
  si dimostra col CUT&PASTE

- Proprietà di sottostruttura ottima con spazio dei sottoproblemi lineare

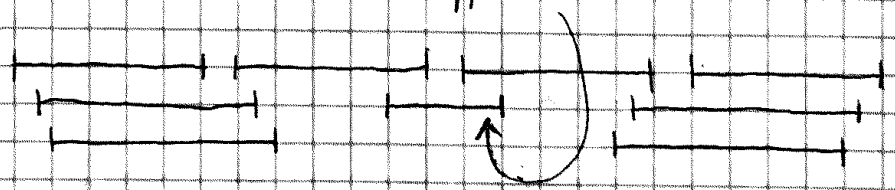
  significa dimostrare che la sol. ottima, che contiene la scelta greedy, contiene una soluzione a l'unico sottoproblema da risolvere dopo aver fatto la scelta greedy

Esercizio: scelte greedy alternative per selezione di attività

- scegli l'attività di durata inferiore
  No; controesempio:

- scegli l'attività che inizia prima
  No; controesempio:

- scegli l'attività col minor numero di sovrapposizioni
  No; controesempio:



- scegli l'attività che inizia per ultima
  Sì, porta ad un alg. greedy corretto → esercizio

Compressione dei dati

compressione < lossless / ~~lossy (e.g. jpeg)~~

File con 100k caratteri ASCII ⟶ 800Kbit

	a	b	c	d	e	f
frequenza (%)	45	13	12	13	9	5

min n° di bit per codificare 6 caratteri è 3: non più 8

a	b	c	d	e	f	
000	001	010	011	100	101	← "codeword"

serve anche una tabella di conversione:

'a'
'b'
'c'
'd'
'e'
'f'

} 6 byte = 48 bit

→ in totale mi servono 300Kbit + 48 bit: risparmio > 50%

decodifica: 001 | 011 | 100 → lookup in posizione 4

↓ b ↓ d ↓ e

In generale:

C = alfabeto dei simboli presenti nel file

baseremo la codifica su C e non su tutto l'universo dei simboli possibili

da determinare:

funzione di encoding $e : C \longrightarrow \{0,1\}^*$

proprietà che e deve avere:

- invertibile $\longrightarrow a \neq b \Rightarrow e(a) \neq e(b)$
- ammettere algoritmi efficienti, come e^{-1}

<u>Fixed-length encoding</u>: associa ad ogni carattere di C una stringa di 0 e 1 della stessa lunghezza

è l'idea più semplice possibile. Nota: ignora totalmente le frequenze

$\rightarrow$ posso usare questa informazione per migliorare le cose?