

Esercizi su Algoritmi Greedy

①

Vediamo una serie di problemi che possono essere risolti con algoritmi greedy. Per ognuno di questi problemi si dovrà

- Impostare un algoritmo che risolva il problema
- Dimostrarne la correttezza
- Valutarne la complessità computazionale (cercate quindi un algoritmo di complessità minima)

Discussione sulle soluzioni greedy:

Questi problemi sono tutti risolvibili con un algoritmi greedy, quindi nella maggior parte dei casi, anche se non tutti, una soluzione consiste in un sottoinsieme dell'input.

$$S \subseteq I$$

• E tipicamente l'analisi del problema consiste in un procedimento del tipo:

- Identificare una sottostruttura ottima

$$S = \{e\} \cup S' \text{ dove } S' \text{ è la soluzione di un sottoproblema}$$

Esempio: Selezione di attività $S_{i,f} = \{(a,b)\} \cup S_{b,i,f}$

ovvero una soluzione ottima nell'intervallo $i-f$ è costituito da un'attività (a,b) con $i \leq a < b \leq f$ e una soluzione ottima per il problema ristretto all'intervallo

②

• Una volta riconosciuta la sottostruttura ottima, è più facile immaginare l'algoritmo e analizzare la correttezza

- Soluzione ottima $S^* = \{e\} \cup S_1^*$

- Soluzione trovata dall'algoritmo $S = \{e\} \cup S_1$

Si può far vedere che la soluzione $S' = \{e\} \cup S_1^*$ è ottima anch'essa.

QUESTO SCHEMA È SOLO INDICATIVO E VA ADATTATO DI VOLTA IN VOLTA AL PROBLEMA

Esercizio 1 File su disco

- Abbiamo dei file di dimensione $d_1, d_2, d_3, \dots, d_n$ e un disco di capacità C
- Trovate un algoritmo di selezione il maggior numero di files che sia possibile mettere nel disco senza superare la capacità.

Esercizio 2 A oppure B

• Abbiamo due liste $A = \{a_1, a_2, \dots, a_{2n}\}$

$B = \{b_1, b_2, \dots, b_{2n}\}$

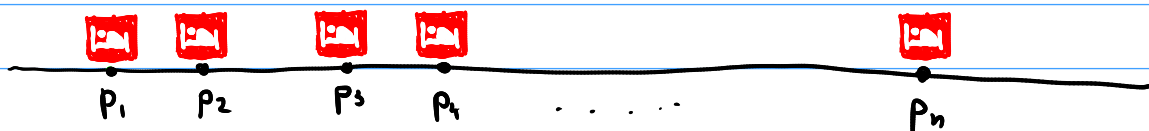
di interi

- Trovate un algoritmo di selezione n elementi di A e n elementi di B tali che
 - ① a_i è scelto se e solo se b_i non è scelto
 - ② la somma degli elementi scelti sia massima

3

Esercizio 3 Chioschi di Gelati

Lungo una spiaggia rettilinea abbiamo degli alberghi alle posizioni p_1, p_2, \dots, p_n



Un imprenditore vuole mettere dei chioschi di gelati sulla spiaggia con il vincolo che ogni albergo deve avere un chiosco entro una distanza di massimo D .

- Trovate un algoritmo che, dati

- p_1, p_2, \dots, p_n già ordinati
- $D \geq 0$

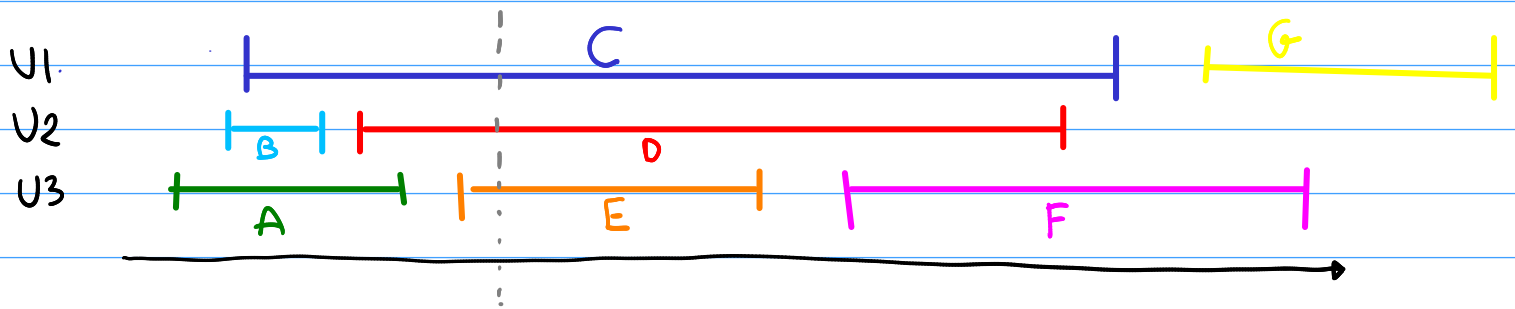
trovi delle posizioni per i chioschi che

- Rispettino i vincoli indicati
- siano il minor numero possibile

Esercizio 4 Assegnazione di attività

- Abbiamo una serie di compiti che deve essere eseguita, e ogni compito è descritto da un intervallo temporale in cui quel compito si svolge (a_i, b_i) con $a_i \leq b_i$ a_i, b_i interi
- Vogliamo eseguire tutti i compiti
- Ogni compito richiede un ufficio per essere svolto
- Non si possono svolgere due compiti che si sovrappongono nello stesso ufficio
- Trovare un algoritmo che assegni i compiti al minor numero di uffici

Esempio



vedete che questi compiti possono essere assegnati a 3 uffici, e vedete che la linea tratteggiata mostra un momento in cui 3 compiti devono essere eseguiti in parallelo. Perciò 3 uffici sono necessari. Dunque questa soluzione è ottima

OSSERVAZIONE

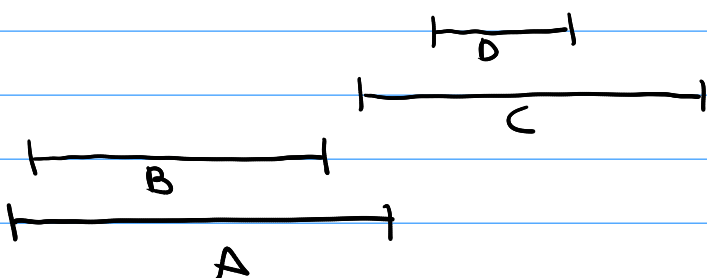
Un'idea potrebbe essere:

- selezionare il maggior numero di attività usando l'algoritmo greedy di selezione delle attività che abbiamo già visto
- assegnare queste attività ad un nuovo ufficio
- ripetere il procedimento sulle attività non assegnate

QUESTO PROCESSO NON PORTA A SOLUZIONE OTTIMA

Lo vediamo in questo esempio:

Input



Soluzione ottenuta dall'approccio descritto

-
-
-

Soluzione ottima

-
-