

Laboratorio Fondamenti di Programmazione 2

30/11/2023

Esercizio 1

Un grafo si dice k -colorabile se è possibile assegnare a ciascun suo nodo un colore tra c_1, \dots, c_k , dove k è un numero intero positivo, in modo tale che non esista nessun arco del grafo i cui vertici siano colorati con lo stesso colore. Intuitivamente, se (u, v) è un arco del grafo, è necessario che i nodi u, v siano colorati con colori diversi.

Scrivere una funzione che preso in input un grafo non orientato G , calcoli una $(D + 1)$ -colorazione, dove D è il massimo grado nel grafo.

Si può dimostrare che questa colorazione esiste sempre (teorema di Brook), in particolare può essere calcolata implementato una strategia greedy (~ usare il primo “colore disponibile” per ogni nodo). Notare che $D + 1$ (il numero di colori) dipende da una proprietà del grafo - da non confondere con il problema della k -colorabilità, che richiede una k -colorazione di un grafo, ma dove k non dipende dal grafo in input. Quest’ultimo problema è “più difficile” dal punto di vista computazionale.

Esercizio 2

Sia M una matrice di interi positivi di dimensione $n \times m$. Siamo interessati a raggiungere l’angolo in basso a destra, di coordinate $(n - 1, m - 1)$, a partire dall’angolo in alto a sinistra, di coordinate $(0, 0)$, muovendoci solo a destra o in basso, attraverso il percorso *di costo minore*. Il costo di un percorso è calcolato come la somma dei valori nelle celle attraversate.

Esempio: Consideriamo questa matrice di dimensione 4×3 :

```
10 30 90
50 30 90
10 90 90
10 10 10
```

Uno dei possibili percorsi per raggiungere $(3, 2)$ da $(0, 0)$ è DDDGGG, con il costo totale di $10 + 30 + 90 + 90 + 90 + 10 = 320$. Il percorso ottimo è GGGDD, con costo totale di $10 + 50 + 10 + 10 + 10 + 10 = 100$.

1. Implementare un algoritmo greedy per risolvere il problema, in particolare scegliendo sempre la direzione *localmente più conveniente*.
2. Applicare la tecnica di programmazione dinamica al problema.

Esercizio 3

Scrivere un programma che presi in input n numeri interi distinti e un numero positivo k , con $0 < k < n$ calcoli la sottosequenza di k elementi di somma massima.

Il programma dovrà stampare la sottosequenza ottima, la sua somma e le posizioni della sequenza originale in cui appaiono gli elementi della sottosequenza ottima.

Esempio. Consideriamo la sequenza 8 13 -50 3 2 9 e $k = 3$. In questo caso, la sottosequenza ottima è composta dagli elementi 8, 13, 9, la sua somma è 30 e la posizione in cui compaiono gli elementi interessati sono 0, 1, 5.

Esercizio 4

Non tutti sono in grado di vedere immediatamente qual è la più equa divisione di monete fra due persone. Dato un sacchetto contenente un massimo di 100 monete, determinare la più equa divisione tra due persone. Ciò significa che la differenza tra il valore delle monete assegnate ad ogni persona dovrebbe essere minimizzato. Si suppone che il valore di una moneta varia da 1 centesimo a 500 centesimi. Non è consentito suddividere una singola moneta.

1. Proporre un approccio greedy.
2. Proporre una soluzione dinamica per il problema.