

Problem Set 3

Algoritmi e Strutture Dati – a.a. 2023/2024

Università di Roma “Tor Vergata”

Prof. Luciano Gualà

Linee guida per la consegna. Gli elaborati dovranno essere consegnati entro domenica 05/05/2024 alle ore 23:59. Si prega di seguire le seguenti indicazioni:

1. le soluzioni dovranno essere consegnate via email a entrambi gli indirizzi `guala@mat.uniroma2.it` e `alessandrostr95@gmail.com`;
2. l'oggetto dell'email dovrà essere “Consegna problem set 3 - 2023/2024”;
3. il file dovrà essere in formato pdf;
4. il nome del file dovrà essere “PS_3_XXX.pdf” dove XXX è il primo cognome (in lower case) in ordine alfabetico dei membri del gruppo;
5. l'email dovrà essere inviata tramite l'indirizzo dello studente XXX presente nel nome del file;
6. inserire nel file nome, cognome, indirizzo email e matricola di tutti i membri del gruppo (meglio ancora se tutti i membri del gruppo sono in indirizzo nell'email della consegna).

Buon lavoro 🍀 😊!

Problema 1 (*i docenti cercano empatia*)

Una cosa che gli studenti di solito non capiscono è quanto è difficile e a volte frustrante correggere esercizi di algoritmi. In particolare, per noi docenti è faticoso capire se un algoritmo proposto da uno studente funziona o meno (figuriamoci poi valutarlo).

Questo esercizio prova a farvi fare quest'esperienza. Vi proponiamo un problema e un algoritmo. Il vostro compito è correggerlo, ovvero capire se è corretto. Ovviamente dovete essere in grado di argomentare la vostra opinione.

Il problema è il seguente: avete in input una *sequenza* di $n \geq 1$ cifre $\mathbf{x} = x_{n-1}x_{n-2}\dots x_0 \in \{0, 1, \dots, b-1\}^n$ ed un intero $k \geq 0$. Questa sequenza può facilmente essere interpretata come il numero in base b di valore

$$(\mathbf{x})_b = x_0b^0 + x_1b^1 + \dots x_{n-1}b^{n-1}$$

Si vogliono eliminare *esattamente* k cifre dalla sequenza, in modo tale che la sequenza risultante $\mathbf{x}' = x'_{n-k-1}x'_{n-k-2}\dots x'_0$ abbia *valore massimo* se interpretata come un numero in base b .

L'algoritmo che vi proponiamo esegue k iterazioni, dove ad ogni iterazione viene rimossa in maniera *greedy* una cifra dalla sequenza secondo il seguente criterio.

Sia x'_{h-1}, \dots, x'_0 la sequenza all'inizio di una generica iterazione. In questa iterazione leggiamo le cifre da sinistra a destra e scegliamo la prima cifra x'_j che è seguita da una cifra x'_{j-1} strettamente più grande. Se tale cifra x'_j esiste rimuoviamo proprio x'_j , altrimenti, rimuoviamo l'ultima cifra x'_0 . Un esempio di esecuzione dell'algoritmo è riportato in Figura 1.

Dimostrate la correttezza dell'algoritmo greedy proposto o, se pensate che non sia corretto, fornite un controesempio.

```
6 5 4 9 6 0 4 3 2
      #
6 5 9 6 0 4 3 2
      #
6 9 6 0 4 3 2
      #
      9 6 0 4 3 2
          #
          9 6 4 3 2
              #
              9 6 4 3
```

Figure 1: Esempio di esecuzione dell'algoritmo su una sequenza di $n = 9$ cifre in base 10 e con $k = 5$.

Problema 2 (*Un gioco con pedine e scacchiera*)

Vi è consegnata una scacchiera che ha 3 righe e n colonne. Ogni cella della scacchiera, diciamo di coordinate (i, j) , ha un punteggio associato $\text{score}(i, j)$. Avete a disposizione un numero illimitato di pedine che potete piazzare sulla scacchiera secondo le seguenti regole.

- Una pedina posizionata sulla cella (i, j) vi fa guadagnare $\text{score}(i, j)$;
- Non potete piazzare pedine su celle adiacenti, né orizzontalmente né verticalmente. Quindi, se piazzare una pedina in posizione (i, j) , dovete lasciare senza pedine le celle (se esistono) $(i, j - 1)$, $(i, j + 1)$, $(i - 1, j)$, $(i + 1, j)$.

Progettate un algoritmo di programmazione dinamica che trovi il massimo punteggio ottenibile per una certa scacchiera. La complessità temporale dell'algoritmo deve essere $O(n)$. Un esempio di istanza è riportato in Figura 2.

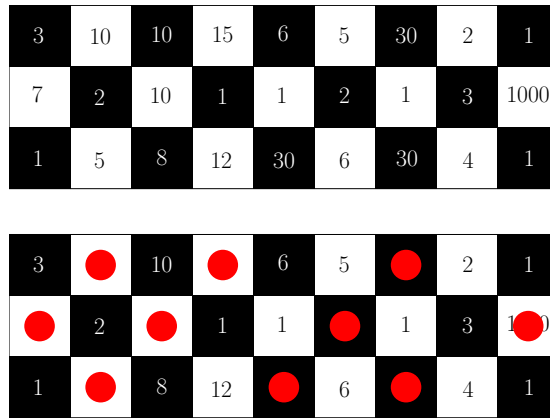


Figure 2: Scacchiera di larghezza $n = 9$. Sotto è riportata una soluzione ammissibile con punteggio 1139. Sarà ottima? Lo saprete solo se progetterete un algoritmo corretto e lo darete in pasto a un calcolatore.

Domanda di approfondimento (opzionale). Che succede se avete anche il vincolo che il numero totale di pedine piazzate non può superare un certo valore Δ ? Riuscite a progettare un algoritmo di programmazione dinamica che in tempo polinomiale calcola il miglior punteggio ottenibile anche per questa generalizzazione del problema?

Problema 3 (*costruire un albero di costo minimo*)

Vi sono dati in input n interi positivi $c_1, \dots, c_n \in \mathbb{N}$ e volete costruire su albero binario che ha tali valori come foglie. In particolare, diremo che un albero binario T è ammissibile se rispetta le seguenti proprietà.

- ogni nodo interno di T ha *esattamente* 2 figli;
- T ha esattamente n foglie.

Ad ogni nodo di un albero ammissibile T è associato un costo. Il costo della foglia i -esima in ordine di visita simmetrica è esattamente il valore c_i . Il costo di un nodo interno è il prodotto tra il massimo costo delle foglie del suo sottoalbero sinistro per il massimo costo delle foglie del suo sottoalbero destro. Il costo dell'albero T è la somma dei costi dei suoi nodi interni. Progettate un algoritmo di programmazione dinamica che calcola un albero ammissibile di costo minimo. Un esempio è dato in Figura 3.

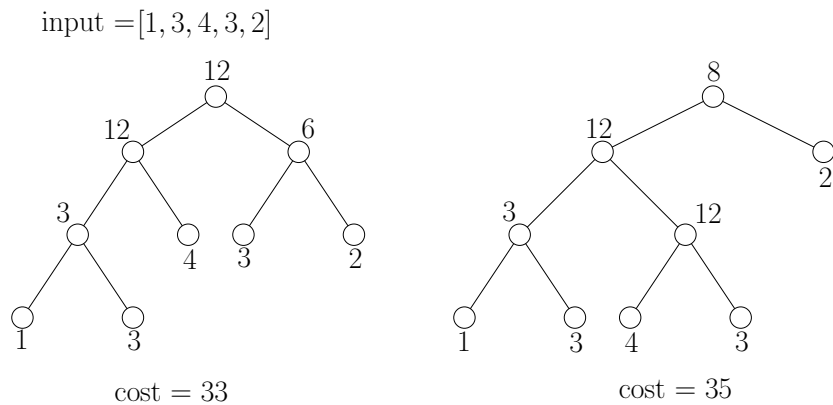


Figure 3: Due alberi binari ammissibili di costo 33 (albero sinistro) e 35 (albero destro), con input 1, 3, 4, 3, 2.