

Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 20/6/2022

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

Avvertenza: Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia dato un insieme S di n numeri interi **distinti**. Gli elementi di S sono memorizzati in parte in un *max-heap* A e in parte in un *min-heap* B , entrambi non vuoti.
Rispondere alle seguenti domande.
 - a. Se tutte le chiavi di A sono *minori* di quelle di B , quanto costa determinare il *minimo* di S ? Quanto costa eliminarlo?
 - b. Se tutte le chiavi di A sono *maggiori* di quelle di B , quanto costa determinare il *minimo* di S ? Quanto costa eliminarlo?
2. Il Prof. O. B. Vious sostiene di aver dimostrato che ISOMORFISMO-DI-GRAFI è riducibile polinomialmente a CLIQUE, e di aver quindi diritto al premio di un milione di dollari messo in palio per la risoluzione del problema del millennio “P=NP?”. L’affermazione del professore è plausibile? Se fosse vera, avrebbe diritto al premio?
3. Dato un grafo orientato G con n vertici ed m archi, quante sono esattamente le operazioni di rilassamento (“relax”) realizzate dall’algoritmo di Dijkstra su G ? Sarebbe possibile ridurle senza alterare la correttezza dell’algoritmo? (Spiegare.)

Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 20/6/2022

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

Avvertenza: Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia T un **albero binario di ricerca** contenente n chiavi intere **distinte**. Sia k una chiave di T . Si consideri il problema di eliminare da T tutte le chiavi *minori* di k .
 - a) Si scriva una procedura in C o in C++ **efficiente** per risolvere il problema proposto. I parametri della procedura sono un albero t di tipo `Tree` e un intero k . **Specificare quale linguaggio è stato usato.**
 - b) Calcolarne e giustificare la complessità.
 - c) Scrivere il tipo in C o in C++ `Tree` ed eventuali altri tipi che sono utilizzati.
2. Sia A un vettore di **interi positivi** di lunghezza n . Si consideri il problema di disporre gli elementi in A in modo che il valore assoluto della differenza tra due elementi adiacenti decresca scorrendo il vettore da sinistra verso destra. Formalmente, per ogni $i \in [1, n-2]$ dovrà valere che

$$|A[i] - A[i+1]| \geq |A[i+1] - A[i+2]|$$

- a. Quali elementi di A dovranno essere spostati in $A[1]$ e $A[2]$ per rispettare la condizione sopra presentata?
- b. Si scriva lo pseudo-codice di una procedura **efficiente** per risolvere il problema descritto. Valutare e giustificare la complessità della procedura proposta.
- c. Si scriva lo pseudo-codice di una procedura **efficiente** per risolvere il problema descritto nel caso in cui la differenza tra il massimo e il minimo di A sia inferiore a n . Valutare e giustificare la complessità della procedura proposta.

Si devono scrivere le eventuali funzioni/procedure ausiliarie utilizzate.

3. a) Si risolva **con il metodo dell'albero di ricorsione** (e niente altro) la seguente ricorrenza:

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + n^3 & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

- b) Si risolva **con il metodo dell'iterazione** (e niente altro) la seguente ricorrenza:

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + 1 & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

Nota - Si ricordi che, per $q \neq 1$, si ha: $\sum_{k=0}^n q^k = (1 - q^{n+1})/(1 - q)$.

4. Sia $G = (V, E)$ un grafo orientato con funzione peso $w : E \rightarrow \mathbb{R}$ e vertici numerati da 1 a n , cioè $V = \{1, 2, \dots, n\}$, e sia k una costante intera con $k \leq n$. Si scriva un algoritmo che, per ogni coppia di vertici $i, j \in V$, determini la lunghezza del cammino minimo tra i e j i cui vertici intermedi non superino k . Si dimostri la correttezza dell'algoritmo proposto e si determini la sua complessità (in funzione di n). Si assuma che i pesi sugli archi siano positivi e che quindi G non contenga cicli negativi.