Algoritmi e Strutture Dati
24 gennaio 2025

Note

1. La leggibilità è un prerequisito: parti difficili da leggere potranno essere ignorate.
2. Quando si presenta un algoritmo è fondamentale spiegare l'idea e motivarne la correttezza.
3. L'efficienza e l'aderenza alla traccia sono criteri di valutazione delle soluzioni proposte.
4. Si consegnano tutti i fogli, con nome, cognome, matricola e l'indicazione *bella copia* o *brutta copia.*

Domande

Domanda A (7 punti) Dare la definizione di max-heap. Dato un insieme S di elementi, memorizzato in parte in un min-heap A e in parte in un max-heap B, entrambi non vuoti, dare un algoritmo `min(A,B)` per trovare il minimo di S nelle due situazioni seguenti:

(a) ogni elemento di A è minore o uguale a ogni elemento di B;

(b) ogni elemento di B è minore o uguale a ogni elemento di A.

In entrambi i casi scrivere lo pseudo-codice e valutare la complessità.

Domanda B (7 punti) Si consideri il problema di selezione di attività compatibili:

(a) Definire il problema.

(b) Descrivere brevemente l'algoritmo ottimo GREEDY-SEL visto in classe.

(c) Fornire un esempio di algoritmo greedy *non* ottimo, motivandone la non ottimalità.

Esercizi

Esercizio 1 (10 punti) Sia dato un array `V[1..n]` i cui valori rappresentano la variazione giornaliera del valore di un titolo azionario. È noto che il titolo è stato prima perdita, con valori sempre negativi, poi ha iniziato a oscillare in giorni *consecutivi* tra valori positivi e negativi, e infine si è stabilizzato su valori positivi (dunque nella sequenza non ci possono essere due giorni positivi seguiti da un negativo). Realizzare un algoritmo *divide et impera* `Split(V)` che individua il giorno in cui il titolo ha iniziato a essere stabile su valori positivi, ovvero il minimo indice $i \in [1, n]$ tale che per ogni $j \geq i$ vale $V[j] > 0$. Se il titolo non si stabilizza su valori positivi, ritornare 0. Ad es., se l'array `V = [-1,-2,2,-1,6,3]` l'indice da tornare sarà `5`, mentre invece per `V=[-1,-2,2,-1,6,-3]` si ritornerà `0`. Fornire lo pseudocodice di `Split(V)`, motivarne la correttezza e individuarne la complessità. Si assuma che non ci siano valori nulli.

Esercizio 2 (8 punti) Date due stringhe $X = \langle x_1, x_2, \ldots, x_m \rangle$ e $Y = \langle y_1, y_2, \ldots, y_n \rangle$, si consideri la seguente quantità $\ell(i, j)$, definita per ogni coppia di valori i, j con $0 \leq i \leq m$ e $0 \leq j \leq n$:

$$\ell(i,j) = \begin{cases} 1 & \text{se } i = 0 \text{ o } j = 0 \\ 3\ell(i, j-1) & \text{se } i, j > 0 \text{ e } x_i = y_j \\ 2\ell(i-1, j-1) - \ell(i-1, j) & \text{se } i, j > 0 \text{ e } x_i \neq y_j. \end{cases}$$

Si vuole calcolare la quantità $q = \max\{\ell(i, j) : 0 \leq i \leq m, 0 \leq j \leq n\}$.

(a) Scrivere un algoritmo bottom-up per il calcolo di q.

(b) Determinare la complessità *esatta* dell'algoritmo, supponendo che le uniche operazioni di costo unitario e non nullo siano i confronti tra caratteri.