

Problem Set 3

docente: Luciano Gualà

Esercizio 1 *(una domanda semplice ma non troppo)*

Dire se può esistere un algoritmo di ordinamento basato su confronti che ordina un insieme di 8 elementi facendo nel caso peggiore al più 15 confronti. Motivare la risposta.

Esercizio 2

Sia T un albero binario di n nodi con radice r in cui ogni nodo ha un valore non negativo associato. La *profondità* di un nodo v è il numero di archi del cammino da v alla radice. I nodi che si incontrano lungo tale cammino (v compreso) sono detti *antenati* di v . Diremo che un nodo v è *generazionalmente profondo* se la sua profondità è strettamente maggiore del valore di un suo antenato di valore minimo.

Si assuma che T è mantenuto attraverso una struttura collegata e che ogni nodo v abbia associato i seguenti campi: puntatori al padre e ai figli ($v.p$, $v.s$, $v.d$) e valore del nodo ($v.val$). Si progetti un algoritmo con complessità temporale $O(n)$ che, preso T , restituisca il numero di nodi generazionalmente profondi di T . Si fornisca lo pseudocodice dettagliato dell'algoritmo.

Esercizio 3

Nella Striscia Di Mezzo si combatte una guerra sanguinaria fra le Forze Del Bene e le Forze Del Male. La Striscia si estende da ovest a est ed è composta da n territori, numerati (da ovest a est) da 1 a n . Il territorio i -esimo confina con il territorio $(i - 1)$ -esimo e $(i + 1)$ -esimo, $i = 2, 3, \dots, n - 1$. Il territorio 1 confina solo con il territorio 2 ed il territorio n confina solo con il territorio $n - 1$. Ogni territorio è controllato da uno dei due schieramenti e, per ogni territorio controllato dalle Forze Del Bene, sono noti il numero di armate che difendono tale territorio. Una possibile situazione nella Striscia Di Mezzo è rappresentata nella seguente figura, dove i territori controllati dalle Forze Del Male sono rappresentati da teschi, mentre per gli altri territori (controllati dalle Forze Del Bene) è indicato il numero di armate presenti sul territorio:



Se le Forze Del Male di un territorio i vogliono conquistare un territorio j controllato dalle Forze Del Bene, devono farsi strada sconfiggendo tutte le armate delle Forze del Bene che si trovano nei territori tra i e j (j compreso). Per ogni territorio j controllato dalle Forze Del Bene, definiamo il *fattore di difesa* δ_j come il numero minimo di armate che le Forze Del Male devono sconfiggere per conquistare il territorio j . Ad esempio, il territorio con indice 1 in figura ha fattore di difesa $\delta_1 = 5$ mentre quelli con indici 8 e 9 hanno fattori di difesa $\delta_8 = 9$ e $\delta_9 = 7$.

Si progetti un algoritmo che, preso in input in intero $k \in \{1, 2, \dots, n\}$, calcola in tempo $O(n + k \log n)$ i k territori meglio difesi dalle Forze del Bene (cioè i k territori con i più alti fattori di difesa).

Suggerimento: è possibile calcolare tutti i fattori di difesa δ_j dei territori controllati dalle Forze Del Bene in tempo $O(n)$.

Esercizio 4

Sia A una matrice $n \times n$ di numeri non negativi. Progettare un algoritmo che, data A , costruisca un *oracolo* (ovvero una struttura dati) che sia in grado di rispondere a *query* (ovvero domande) del seguente tipo:

- **query1**(i, j, b, h): dati quattro interi i, j, b, h , restituire la somma degli elementi della sottomatrice di A delimitata dai quattro elementi $A[i, j], A[i, j + b - 1], A[i + h - 1, j], A[i + h - 1, j + b - 1]$.
- **query2**(x): dato un valore x , restituire, se esistono, una coppia di indici (i, j) tale che la somma degli elementi della sottomatrice delimitata dai quattro elementi $A[1, 1], A[1, j], A[i, 1], A[i, j]$ è uguale a x .

L'oracolo deve avere dimensione $O(n^2)$ e l'algoritmo che lo costruisce deve avere complessità temporale lineare (nella dimensione dell'istanza), ovvero $O(n^2)$. Per quanto riguarda la complessità delle query, **query1** e **query2** devono poter essere risposte rispettivamente in tempo costante e tempo $O(n)$.

Un'ulteriore domanda: sempre tenendo la dimensione dell'oracolo a $O(n^2)$ e il tempo di risposta costante a **query1**, concedendosi tempo di costruzione $O(n^2 \log n)$, è possibile portare il tempo richiesto per la **query2** da $O(n)$ a $O(\log n)$?