

Cognome:..... Nome:..... Matr:.....

Esercizio 1 [11 punti] Si consideri il problema del mantenimento di insiemi disgiunti attraverso una struttura dati Union-Find.

1. Dire quale delle seguenti affermazioni è vera:

- Nella QuickUnion con euristica *union by size* ogni insieme è rappresentato con un albero di altezza 1, in modo che sia l'operazione di **find** che di **union** richiedano tempo logaritmico.
- Usando la struttura dati QuickUnion con euristica *union by size*, ogni operazione di **find** ha costo ammortizzato $O(\log n)$, dove n è il numero di **makeSet**. Eppure una singola operazione di **find** nel caso peggiore può costare anche $\Theta(n)$.
- Usando la struttura dati QuickFind con euristica *union by size*, ogni sequenza di n **makeSet**, $n - 1$ **union** e m **find**, richiede nel caso peggiore tempo $O(m + n \log n)$.
- Ogni struttura dati, per eseguire una sequenza di n **makeSet**, $n - 1$ **union** e m **find**, deve impiegare nel caso peggiore tempo $\Omega(m + n)$.
- Si assuma di implementare l'algoritmo di Kruskal usando una struttura dati QuickFind con euristica *union by size*. Si assuma inoltre di avere già gli archi del grafo ordinati in ordine non decrescente rispetto al loro peso. Allora l'esecuzione dell'algoritmo di Kruskal ha complessità temporale $O(m + n \log n)$.

2. Si consideri la struttura dati QuickUnion con euristica *union by size*. Si mostri una sequenza di operazioni di n **makeSet** e $n - 1$ **union** in cui l'albero ottenuto abbia altezza $\Theta(\log n)$. (Max 5 righe.)

Esercizio 2 [11 punti] Si consideri il problema dell'Interval Partitioning (IP).

1. Si definisca formalmente il problema di IP. (Max 5 righe.)

2. Si definisca il concetto di *depth* di un'istanza di IP e si discuta perché è importante per analizzare l'algoritmo greedy che risolve IP. (Max 5 righe.)

Esercizio 3 [11 punti] Siete appena tornati da un viaggio nel tempo e mentre eravate nel futuro avete preso i numeri vincenti della lotteria settimanale. Così ora avete n numeri vincenti, per ognuna delle prossime n settimane. Conoscete anche il valore della vincita v_i per ogni settimana i .

Ora, il problema è che la polizia sta cercando i viaggiatori nel tempo, quindi dovete cercare di non destare sospetti. Così decidete di giocare alla lotteria con il seguente vincolo: fra una vincita e l'altra devono esserci *almeno* 3 settimane in cui non vincete. Progettate un algoritmo di programmazione dinamica che calcola il valore massimo che riuscirete a guadagnare nelle prossime n settimane. Si discuta la complessità temporale dell'algoritmo proposto.