

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

Esercizio 1 [11 punti] Si consideri il problema del calcolo del minimo albero ricoprente (*Minimum Spanning Tree problem*) che prende in input un grafo non orientato e pesato $G = (V, E, w)$.

1. Dire quale delle seguenti affermazioni è vera:

- Sia T un MST di G e sia f un arco non di T . Allora f è l'arco di peso massimo in tutti i cicli che lo contengono.
- Sia v un nodo qualsiasi. L'arco più leggero incidente a v fa parte sempre di un qualche MST di G .
- Se tutti i pesi di G sono uguali, allora ogni albero dei cammini minimi di G rispetto a una qualsiasi sorgente s è anche un MST di G .
- Se per ogni arco e vale $w(e) \in \{1, 2\}$, allora l'algoritmo di Prim applicato su un nodo iniziale s calcola un MST che è anche un albero dei cammini minimi con sorgente s .
- Se G è completo allora l'algoritmo di Kruskal ha la stessa complessità asintotica sia se usa per la struttura Union-Find la *QuickFind* con o senza euristica *union by size*.

2. Si consideri la seguente affermazione e si dica se è (sempre) vera o (se può essere) falsa, argomentando la risposta. (Max 5 righe.)

Claim: Sia $G = (V, E, w)$ un grafo non orientato e pesato. Sia T un MST di G e sia e un arco di T . Si consideri il grafo $G' = (V, E, w')$ ottenuto da G abbassando il peso dell'arco e a un valore $w'(e) < w(e)$. Allora T è un MST anche di G' .

Esercizio 2 [11 punti] Si consideri il problema del calcolo del taglio di capacità minima - *Minimum cut problem*.

1. Si definisca formalmente il problema. (Max 5 righe.)
2. Si descriva come è possibile, dato un flusso massimo, calcolare in tempo lineare un taglio di capacità minima. (Max 5 righe.)
3. Si discuta in modo conciso e preciso la correttezza dell'algoritmo fornito nel punto precedente. (Max 5 righe.)

Esercizio 3 [11 punti] Dovete progettare un algoritmo per giocare a *Tube Invaders*, un nuovo videogioco messo in commercio proprio il giorno dello scritto di ASD. *Tube invaders* è un gioco semplice in cui voi controllate un'astronave che avanza dentro un tubo. Il tubo è diviso in n segmenti, corrispondenti alle n battaglie – numerate da 1 a n – che dovete affrontare. Nella battaglia i , sapete che troverete una flotta di a_i alieni. Per affrontare le battaglie, avete a disposizione un'arma. L'arma, quando non la usate, accumula energia. L'energia rappresenta il massimo numero di alieni che potete distruggere quando decidete di usarla in battaglia. Subito dopo aver usato l'arma, l'energia della vostra arma diventa 1. Anche all'inizio, nella prima battaglia l'energia è 1, mentre l'energia raddoppia ad ogni battaglia in cui non la usate.

Progettate un algoritmo di programmazione dinamica che calcoli il massimo numero di alieni che potete distruggere in una generica istanza di *Tube Invaders*. Si discuta la complessità temporale dell'algoritmo proposto.