

Algoritmi e Strutture Dati (modulo II) - testo prova scritta 28/09/2022  
docenti: Luciano Gualà & Andrea Clementi

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

**Esercizio 1 [11 punti]** Si consideri un grafo non orientato e connesso  $G = (V, E)$  di  $n$  nodi dove ad ogni arco  $e \in E$  è associato un peso  $w(e)$ .

1. Dire quale delle seguenti affermazioni è vera:

- Se i pesi degli archi di  $G$  sono distinti, l'arco di peso minimo appartiene sempre all'MST  $T$  di  $G$  mentre l'arco di peso massimo non appartiene mai a  $T$ .
- Sia  $C$  un ciclo di  $G$  ed  $e$  l'arco più leggero di  $C$ . Allora esiste sempre un MST di  $G$  che contiene  $e$ .
- Sia  $T$  un MST di  $G$  e siano  $u$  e  $v$  due nodi qualsiasi. Sia  $e$  un arco che appartiene all'unico cammino in  $T$  fra  $u$  e  $v$ . Allora se esiste l'arco  $f = (u, v)$  in  $G$  deve essere  $w(f) \geq w(e)$ .
- Se i pesi degli archi di  $G$  sono distinti, allora l'algoritmo di Prim e quello di Kruskal restituiscono lo stesso MST, ma solo scegliendo in modo opportuno la sorgente  $s$  da cui lanciare l'algoritmo di Prim.
- Se i pesi degli archi di  $G$  sono distinti e interi, gli algoritmi di Prim e di Dijkstra lanciati con la stessa sorgente  $s$  calcolano lo stesso albero.

2. Immaginate di implementare l'algoritmo di Kruskal con un'altra struttura dati Union-Find i cui costi delle operazioni sono: la **MakeSet** e l'operazione di **Union** hanno costo costante, mentre la **Find** costa  $O(\log \log n)$ . Quale sarebbe la complessità dell'algoritmo di Kruskal? Giustificate la risposta. (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]** State preparando la campagna elettorale in vista delle elezioni e volete pianificare dei comizi per aumentare la vostra chance di vincere. Davanti a voi avete  $n$  giorni ma gli esperti dicono che non tutti i giorni garantiscono la stessa visibilità. In particolare, sapete che fare un comizio il giorno  $i$  vi farà guadagnare  $v_i$  voti. Regole di fairness impongono che non potete fare comizi per due giorni consecutivi. Inoltre, i soldi a vostra disposizione per organizzare gli eventi sono limitati e il numero di comizi deve essere al più  $B$ .

Progettate un algoritmo di programmazione dinamica che calcoli il numero massimo di voti che potete guadagnare.

*Una semplificazione:* Un sottocaso più semplice del problema è lo scenario in cui siete schifosamente ricchi e quindi non avete vincoli sul numero di comizi che potete organizzare (ma il vincolo di fairness resta). Se non riuscite a progettare un algoritmo per il caso generale potete risolvere questo sottocaso (che però non dà diritto al punteggio pieno).