

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

Esercizio 1 [11 punti] Si consideri un grafo connesso non orientato e pesato $G = (V, E, c)$ con n nodi ed m archi per il quale si vuole calcolare un *Minimum Spanning Tree (MST)*.

1. Dire quale delle seguenti affermazioni è vera:

- Se ci sono archi di peso negativo posso usare l'algoritmo di Bellman-Ford per calcolare un MST di G .
- L'albero restituito dall'algoritmo di Kruskal non contiene mai l'arco di peso massimo di G .
- L'albero restituito dall'algoritmo di Kruskal ha la proprietà che tutti gli archi che non fanno parte della soluzione hanno un peso strettamente maggiore dell'arco di peso minimo di G .
- L'albero restituito dall'algoritmo di Prim invocato su una sorgente s è anche un albero dei cammini minimi di G rispetto alla stessa sorgente s .
- Gli alberi restituiti dall'algoritmo di Kruskal e dall'algoritmo di Prim potrebbero non essere uguali. Ma in questo caso il grafo contiene più archi dello stesso peso.

2. Si consideri il caso in cui ogni arco e di G ha un costo $c_e \in \{1, 2\}$, e si assuma che ci siano almeno 3 archi di costo 1 in G . Si consideri la seguente affermazione: *Il costo dell'MST di G è al più $2n - 4$* . Dire se l'affermazione è vera o falsa motivando la risposta (Max 5 righe.)

Esercizio 2 [11 punti]

1. Si definisca formalmente il problema del load balancing. (Max 5 righe.)

2. Per l'analisi dell'algoritmo di 2-approssimazione per il problema del load balancing si usano due lemmi tecnici che forniscono dei lower bound al valore della soluzione ottima. Si enuncino formalmente i due lemmi e si descriva brevemente che ruolo giocano nell'analisi del rapporto di approssimazione. (Max 10 righe.)

Esercizio 3 [11 punti] La sessione di esami comincerà domani. Ci sono n giorni davanti a te e ogni giorno c'è un esame e tu ancora non hai cominciato a studiare. Ma hai intenzione di rimetterti in riga. Meglio tardi che mai, del resto, no?

Gli esami non sono tutti uguali. Per ogni i , sai che l'esame che si tiene il giorno i è relativo ad un insegnamento da c_i crediti e sai che per prepararlo ti occorreranno almeno g_i giorni di studio consecutivi. Hai le tue modalità. Sai concentrarti su un solo esame alla volta: studi per un esame, lo sostieni e solo dal giorno dopo averlo sostenuto cominci a pensare e a studiare per il prossimo esame (il giorno in cui fai un esame esci con il tuo miglior amico Walter Bianchi per festeggiare).

Progetta un algoritmo di programmazione dinamica che ti suggerisca il massimo numero di crediti che puoi acquisire in questa sessione.

Bonus (per il punteggio pieno): Da quando hai saputo che il matematico Paul Erdős faceva ampiamente uso di anfetamine per essere scientificamente più produttivo, ti sei procurato dal tuo amico Walter una pillolina che, ti ha assicurato, può essenzialmente dimezzare i giorni di studio necessari per superare un esame. In particolare, se prendi la pillola con lo scopo di preparare l'esame del giorno i , il numero di giorni necessari per preparare l'esame diventa $\lceil \frac{g_i}{2} \rceil$. Hai una sola pillola e quindi puoi usare questo trucco una sola volta. Sai progettare un algoritmo in grado di calcolare il numero di crediti massimi ottenibili nella sessione anche in questo caso?