Algoritmi e Strutture Dati (modulo II) - testo prova scritta 26/06/2025

docente: Luciano Gualà

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

Esercizio 1 [11 punti] Si consideri il problema del calcolo del massimo flusso in una rete $G = (V, E, s, t, c)$ con sorgente s e pozzo t e capacità $c(e)$ per ogni arco $e \in E$.

1. Dire quale delle seguenti affermazioni è vera:

- dato un flusso f di G e un st-taglio (A, B), il valore del flusso $v(f)$ è sempre uguale alla capacità del taglio $cap(A, B)$
- L'algoritmo di Ford-Furkerson ha una complessità che in generale può essere esponenziale nella dimensione dell'istanza, ma è sempre polinomiale quando le capacità degli archi sono valori interi.
- Dato un flusso f, se nella rete residua G_f non c'è alcun cammino da s a t, allora f è un flusso massimo per G.
- Sia f un flusso e sia e un arco di un cammino P da s a t nella rete residua G_f. Allora è sempre possibile usare il cammino aumentante P per aumentare il flusso f di $c_f(e)$, dove $c_f(e)$ è il peso di e nella rete residua G_f.
- Sia f un flusso tale che, nella rete residua G_f, s e t sono separati. Sia B l'insieme di tutti e soli i nodi che possono raggiungere t. Allora $(V \setminus B, B)$ è un taglio minimo di G.

2. Si consideri una rete di flusso $G = (V, E, s, t, c)$ di n nodi in cui ogni nodo ha grado entrante al più 3 (mentre il grado uscente di un nodo può essere anche $\Theta(n)$) e la capacità di ogni arco $e \in E$ è un numero intero non più grande di n^2. Si derivi una delimitazione superiore (quanto più stretta possibile) alla complessità temporale dell'algoritmo di Ford-Fulkerson sulla rete G. Si può affermare che in questo caso l'algoritmo è garantito avere complessità polinomiale? (Max 5 righe.)

Esercizio 2 [11 punti] Si consideri il problema del calcolo del *Minimum Spanning Tree (MST)*.

1. Si definisca formalmente il problema. (Max 5 righe.)
2. Si enunci formalmente la proprietà del taglio (*cut property*). (Max 5 righe.)
3. Si discuta in modo conciso e preciso come è possibile usare la proprietà del taglio per dimostrare la correttezza dell'algoritmo di Prim. (Max 5 righe.)

Esercizio 3 [11 punti] Vicino la vostra università ha aperto un nuovo fast food, il *BurgerGata*. Come lancio pubblicitario, i titolari hanno organizzato un'iniziativa denominata *Menu & Coupon* che sarà attiva per i prossimi n giorni, a pranzo. Potete trovare i dettagli sul sito di BurgerGata. Per ogni giorno $i = 1, \ldots, n$, il locale mette a disposizione un menù fisso al prezzo di p_i euro e un coupon di valore c_i, che può essere utilizzato per intero e in un'unica soluzione solo in un giorno successivo. I coupon non sono cumulabili e ogni volta che un coupon è utilizzato viene disattivato. In particolare, se l'ultima volta che avete mangiato al BurgerGata è stato il giorno i e decidete di tornarci il giorno $j > i$, allora lo *sconto effettivo* che ottenete è di $\min\{p_j, c_i\}$ euro.[1] In compenso vi mettete in tasca il coupon di valore c_j che potete utilizzare la prossima volta.

Il vostro obiettivo è quello di decidere i giorni in cui andare da BurgerGata in modo da massimizzare lo sconto totale effettivo, ovvero la somma degli sconti effettivi ottenuti nei singoli pranzi. Progettate un algoritmo di programmazione dinamica che calcoli il massimo sconto totale effettivo che potete ottenere dall'iniziativa. Si discuta la complessità temporale dell'algoritmo proposto.

[1] Il che vuol dire che se per esempio avete un coupon di valore $c_i = 30$ e lo usate un giorno in cui il menù ha costo $p_j = 10$ essenzialmente bruciate 20 euro del valore del coupon.