Algoritmi e Strutture Dati (modulo II)

Testo della prova scritta del 10 luglio 2017

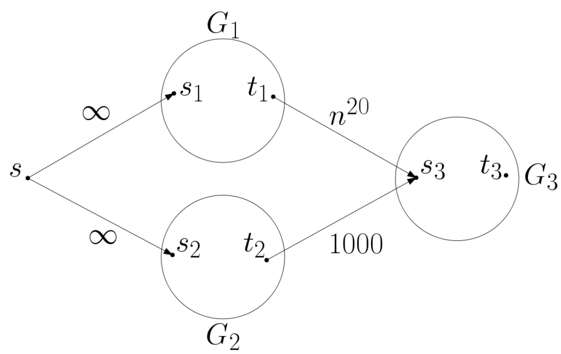
docente: Gualà

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

Esercizio 1 È il 31 luglio del 2756 e avete appena sostenuto l'ultimo esame della sessione estiva e siete finalmente in vacanza. Siete sul pianeta Terra e domani partirete per il pianeta Salentone, un piccolo paese colonizzato da pugliesi dove si va al mare e si mangiano i pasticciotti alla crema più buoni della galassia. Viaggerete con la vostra astronave attraversando lo spazio che può essere descritto come un grafo diretto e pesato $G = (V, E, w)$, dove ogni nodo rappresenta un pianeta, e dove ogni arco (u, v) rappresenta una rotta spaziale percorribile che va dal pianeta u al pianeta v. Il peso $w(e)$ di un arco e indica la quantità di carburante necessario per percorrere la rotta e. Farete bene a non restare senza carburante durante una rotta se non volete rovinarvi la vacanza. La vostra astronave ha a disposizione inizialmente una quantità di carburante pari a Δ, e su ogni pianeta v c'è un distributore che può erogare fino a $\delta(v)$ carburante. I serbatoi di carburante nel 2756 sono realizzati con una tecnologia futuristica usa e getta che ha le seguenti caratteristiche: (i) il serbatoio ha capacità (potenzialmente) infinita, ovvero potete caricarlo senza limiti di capienza; e (ii) può essere ricaricato una sola volta (e quindi potete fare rifornimento ad al più un solo distributore).

Progettate un algoritmo il più efficiente possibile che calcola un modo, se c'è, per andare dalla Terra a Salentone.

Esercizio 2 Un grafo diretto G con capacità sugli archi è composto da un nodo s e da tre grafi G_1, G_2 e G_3, interconnessi come in figura, dove ogni grafo G_i, $i = 1, 2, 3$, è un grafo diretto con capacità intere sugli archi. I tre grafi G_1, G_2 e G_3 hanno n nodi ciascuno ma hanno un diverso numero di archi: G_1 e G_3 hanno $\Theta(n)$ archi mentre G_2 è più denso e ha $\Theta(n^2)$ archi. Dare un algoritmo, il più efficiente possibile, che trova il flusso massimo da s a t_3.



Si riporta di seguito una tabella con i tre algoritmi per il maxflow presentati a lezione e le relative complessità, dove E indica il numero di archi e V il numero di nodi del grafo su cui si applica l'algoritmo e C è una delimitazione superiore (upper bound) al massimo flusso:

Ford–Fulkerson (scegliendo un cammino aumentante qualsiasi)	$C \cdot E$
Δ-scaling	$E^2 \cdot \log C$
Edmonds–Karp (F-F scegliendo un cammino aumentante di lunghezza minima)	$E^2 \cdot V$