Problem Set 4

docente: Luciano Gualà

Esercizio 1 *(una domanda di teoria dei grafi)*
Sia $G = (V, E)$ un grafo non orientato di n nodi. Un *matching* di G è un sottoinsieme $F \subseteq E$ di archi che non condividono nessun vertice; più formalmente, non esistono tre nodi u, v, w tale che $(u, v) \in F$ e $(v, w) \in F$. Un matching F si dice *perfetto* se $|F| = n/2$. Quindi un matching perfetto può esistere solo se G ha un numero pari di nodi e in un tale matching tutti i nodi hanno esattamente un arco di F incidente. Inoltre, G è detto *planare* se è possibile disegnarlo su un piano in modo che gli archi non si intersecano.

Si dimostri che per ogni intero $M \geq 1$ esiste un grafo planare G il cui numero di matching perfetti è esattamente M.

Esercizio 2
Sia $G = (V, E)$ un grafo diretto e non pesato, e sia $p \in V$ un nodo di G. Dati due nodi qualsiasi $u, v \in V$, si definisce *p-cammino da u a v* un cammino in G (anche non semplice) da u a v che attraversa il nodo p. Progettare un *oracolo* (una struttura dati) in grado di rispondere a *query* (domande) del tipo: Quale è la lunghezza del più corto p-cammino da un nodo x a un nodo y? Il tempo di costruzione dell'oracolo deve essere lineare nella dimensione del grafo mentre quello per rispondere ad una generica query deve essere costante. L'oracolo, inoltre, deve avere dimenione $O(|V|)$.

Esercizio 3 (*recuperare il tesoro di Barbablù*)
John Steam della compagnia *Oriental Steam Navigation* decide di organizzare una spedizione di recupero del tesoro del Pirata Barbablù, custodito nel relitto del galeone del pirata, affondato al largo di Gobal, che si trova adagiato su un fianco a diversi metri di profondità. L'unico punto di accesso al relitto è uno squarcio sulla fiancata, in corrispondenza di una specifica cabina. Nel galeone sono presenti cabine e corridoi che le collegano. Tutti i corridoi sono totalmente sommersi dall'acqua a causa della rottura degli oblo mentre in alcune delle cabine sono rimaste delle sacche d'aria. A causa degli spazi angusti non è possibile, per i sommozzatori, esplorare la nave con le bombole d'aria; sono quindi costretti a nuotare in apnea, sfruttando le sacche d'aria presenti nel tragitto per respirare.

Prima di procedere con le operazioni di recupero ti viene commissionata la realizzazione di un algoritmo in grado di individuare il percorso più breve all'interno del galeone che permetta ai sommozzatori di raggiungere la cabina con il tesoro a partire dall'apertura. In alcune cabine sono presenti sacche d'aria che possono essere usate per respirare. Un sommozzatore riesce a nuotare senza aria per Δ metri al massimo prima di dover riprendere fiato.

La mappa del galeone è rappresentata attraverso un grafo pesato $G = (V, E, w)$, dove i nodi rappresentano le cabine e gli archi i corridoi che le collegano. Il peso di ogni arco rappresenta la lunghezza del corridoio. In Figura 1 è mostrato un possibile scenario. La cabina di ingresso è il nodo s, mentre la cabina del tesoro è rappresentata dal nodo t. Le cabine con la sacca d'aria sono in azzurro, mentre quelle senza sacca sono bianche. A fianco di ogni corridoio è segnata la sua lunghezza in metri. L'esempio, per $\Delta = 6$, ammette due soluzioni, la migliore delle quali ha lunghezza 7.

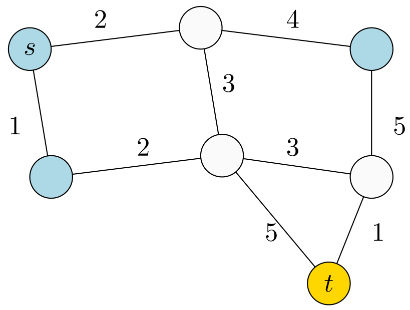

Figura 1: Il galeone con il tesoro di Barbablù, rappresentato come un grafo. La cabina in cui si può entrare è il nodo s, quella con il tesoro è il nodo t.

Le cabine della nave sono n e sono collegate tra loro da m corridoi. Di ogni cabina/nodo si conosce l'eventuale presenza di aria. Si assuma che la cabina del tesoro abbia sempre una sacca d'aria, che consente al sommozzatore di recuperare il tesoro (ripercorrendo al contrario la stessa strada fatta per raggiungere il tesoro).

L'esercizio consiste in due punti.

(a) Si consideri l'algoritmo seguente che, presa la mappa del galeone, restitisce `true` o `false` a seconda dell'esistenza o meno un percorso che consente di recuperare il tesoro. E' un algoritmo corretto? Si argomenti la risposta e si studi la complessità temporale dell'algoritmo.

Algorithm 1: Visita(u, δ)

Nota: Il grafo $G = (V, E, w)$, i nodi che contengono le sacche d'aria, il valore Δ ed i nodi s e t sono informazioni globali.

Nota: L'invocazione iniziale dell'algoritmo è Visita(s, Δ)

Pre. : color(u) = `white` $\forall u \in V$

if $u = t$ **then**
    return `true`

if *u contiene una sacca d'aria* **then**
    $\delta \leftarrow \Delta$

color(u) $\leftarrow$ `black`
foreach $(u, v) \in E$ **do**
    if $w(u,v) \leq \delta$ *and color*(u) = `white` *and* $Visita(v, \delta - w(u,v))$ **then**
        return `true`
color(u) $\leftarrow$ `white`

return `false`

(b) Progettare un algoritmo (più efficiente possibile) che calcoli il percorso più breve che permette ad un sommozzatore di partire dalla cabina s con l'apertura e di raggiungere il tesoro (nodo t), in apnea, sfruttando le eventuali sacche d'aria trovate nel percorso.

Esercizio 4

Sia $G = (V, E)$ un grafo con pesi positivi sugli archi ed $e \in E$ un arco di G. Progettare un algoritmo in grado di determinare in tempo lineare (nella dimensione di G) se esiste un MST di G che contiene l'arco e.