Qualche altro esercizio

Problema 1

Progettare un algoritmo che, dato un grafo diretto aciclico (DAG) $G = (V, E)$, due nodi $s, t \in V$, calcola in tempo lineare nella dimensione del grafo il la lunghezza del cammino (semplice) più lungo da s a t.
Suggerimento: Si usi la programmazione dinamica.

Problema 2

Sia dato un grafo diretto e pesato $G = (V, E, w)$ dove w è una funzione di pesatura che associa a ogni arco $e \in E$ un valore reale non negativo $w(e)$. Si assuma che i nodi V di G siano partizionati in due insiemi (disgiunti) V_R di nodi rossi e V_B di nodi blue. Chiaramente $V_R \cup V_B = V$. Si progetti un algoritmo che, presi anche due nodi s e t, calcoli un cammino minino *alternato* da s a t. Un cammino alternato è un cammino in cui, se si guardano i nodi in ordine da s verso t, i colori dei nodi si alternano a ogni passo, ovvero per ogni nodo v diverso da t, il colore del nodo successivo nel cammino a v è diverso dal colore di v.

Problema 3

Sia dato un grafo diretto e pesato $G = (V, E, w)$ dove w è una funzione di pesatura che associa a ogni arco $e \in E$ un valore reale non negativo $w(e)$. Si assuma che gli archi E di G siano partizionati in due insiemi (disgiunti) E_R di archi rossi e E_B di archi blue. Chiaramente $E_R \cup E_B = E$. Si progetti un algoritmo che, presi anche due nodi s e t, calcoli un cammino minino *alternato* da s a t. Un cammino alternato è un cammino in cui, se si guardano gli archi in ordine da s verso t, i colori di tali archi si alternano a ogni passo, ovvero per ogni arco e escluso quello entrante in t, il colore dell'arco successivo a e è diverso dal colore di e.
Suggerimento: Si può ridurre questo problema al problema precedente.

Problema 4

Sia $G = (V, E)$ un grafo con pesi positivi sugli archi tale che $V = \{s, t\} \cup V_1 \cup V_2 \cup \cdots \cup V_k$ in cui tutti i precedenti insiemi sono a due a due disgiunti. Progettare un algoritmo in grado di trovare il più breve cammino π tra s e

t passante per almeno un nodo di ogni insieme V_i, in ordine. L'algoritmo deve avere complessità $O(k(|E| + |V| \log |V|))$.
Nota: Non è necessario che i nodi degli insiemi V_i siano consecutivi in π. I sottocammini tra nodi di insiemi consecutivi (siano essi V_i e V_{i+1}) possono attraversare qualsiasi vertice del grafo (anche vertici di insiemi V_j con $j < i$ o $j > i + 1$).

Problema 5

Dato un grafo $G = (V, E)$ connesso e con pesi interi sugli archi nell'intervallo $[1, k]$, progettare un algoritmo in grado di calcolare un albero dei cammini minimi di G in tempo $O(k|E|)$.

Problema 6

Sia dato un grafo diretto e pesato $G = (V, E, w)$ dove w è una funzione di pesatura che associa a ogni arco $e \in E$ un valore reale non negativo $w(e)$. Dato un nodo s, si definisce *arborescenza* di G con radice s un albero ricoprente radicato in s tale che tutti gli archi sono orientati dai padri verso i figli. Quindi in un'arborescenza tutti i nodi tranne la radice hanno grado entrante 1, e c'è un cammino diretto dalla radice verso ogni altro nodo. Il peso di un'arborescenza è definito come la somma dei pesi dei suoi archi. Siamo interessati a trovare un'arborescenza di costo minimo. Poiché il problema sembra molto simile a quello del minimo albero di copertura su grafi diretti invece che non diretti, la domanda è: è possibile applicare l'algoritmo di Prim (lanciato con radice s) per trovare un'arborescenza di costo minimo? Trovare un controesempio che mostra che ciò non è possibile. Quale proprietà in questo caso non vale che inficia la correttezza dell'algoritmo di Prim?