Corso di Progettazione di algoritmi

Esercizi 4

ESERCIZIO 1

Sia $G = (V, E)$ un qualsiasi grafo orientato con pesi sugli archi, pesi che possono essere anche negativi ma in cui non sono presenti cicli di peso negativo.

- Dimostrare che l'algoritmo di Dijkstra su grafi di questo tipo non calcola necessariamente i cammini di costo minimo tra la sorgente e gli altri nodi del grafo.

- Per il calcolo dei cammini di costo minimo in G si suggerisce il seguente algoritmo:
  Sia M il costo minimo tra i costi degli archi di G. Modifichiamo i pesi degli archi di G sommando a ciascuno di questi l'intero $|M|$ abbastanza grande da renderli tutti positivi. Al grafo che si ottiene G' (che ha pesi positivi) applichiamo l'algoritmo di Dijkstra.
  I cammini minimi che vengono così calcolati sono anche cammini minimi per il grafo originale G? Motivare la risposta.

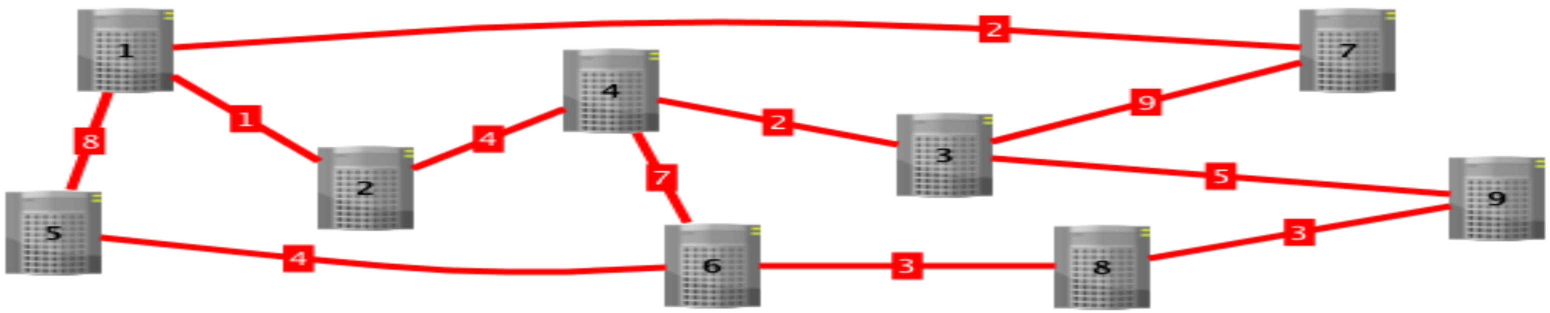
ESERCIZIO 2

Dato un grafo orientato G con pesi positivi sugli archi ed un nodo s di G, l'algoritmo di Dijkstra calcola l'albero dei cammini minimi da s ad un qualsiasi altro nodo di G che è raggiungibile da s. In generale, tra il nodo s e un altro nodo u di G può esserci più di un cammino minimo.

- Descrivere un algoritmo che calcoli per ogni nodo u il numero di tutti i possibili cammini di peso minimo da s a u.
- Discutere la complessità dell'algoritmo.

ESERCIZIO 3

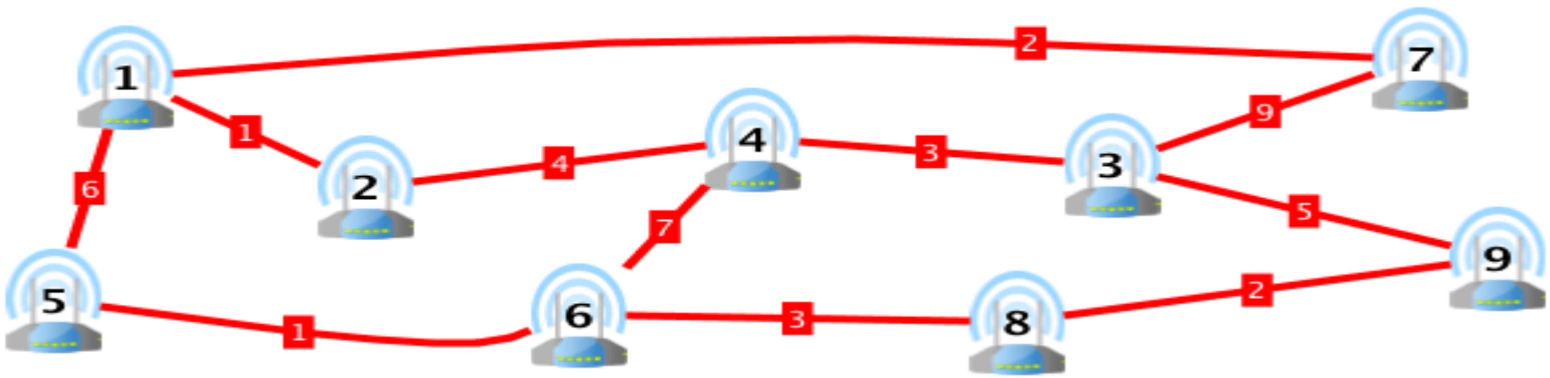
- Consideriamo una rete di comunicazione in cui ogni nodo è un router e se due nodi sono collegati direttamente da una linea di comunicazione (bidirezionale) questa ha una certa capacità espressa tramite un numero intero. Più è alta la capacità di una linea e più è grande la quantità di bit che possono essere trasmessi nell'unità di tempo. Non tutte le coppie di nodi sono collegati direttamente però possono essere connessi passando per altri nodi, cioè tramite un cammino. La capacità di un cammino è la minima capacità delle sue linee di comunicazione. Ad esempio nelle rete qui sotto il cammino che passa per i nodi 1,7,3,9 ha capacità 2 mentre il cammino 1,5,6,8,9 ha capacità 3.



- Dare lo pseudo-codice di un algoritmo efficiente che data una rete e due nodi A e B, trova un cammino di massima capacità tra A e B. Discutere la correttezza e la complessità dell'algoritmo proposto. *Suggerimento: modificare opportunamente l'algoritmo di Dijkstra.*

ESERCIZIO 4

- Consideriamo una rete di comunicazione in cui ogni nodo è una stazione radio e se due nodi possono comunicare direttamente diciamo che sono collegati da una linea di comunicazione (bidirezionale) e questa ha un *livello d'interferenza* espresso tramite un numero intero. Più il livello d'interferenza è alto e più la comunicazione lungo quella linea è disturbata. Non tutte le coppie di nodi sono collegate direttamente però possono essere connesse passando per altri nodi, cioè tramite un cammino. Il livello d'interferenza di un cammino è il massimo livello d'interferenza delle linee di comunicazione del cammino. Ad esempio, nelle rete qui sotto il cammino che passa per i nodi $1, 5, 6, 8, 9$ ha livello d'interferenza 6 mentre il cammino $1, 2, 4, 3, 9$ ha livello d'interferenza 5.

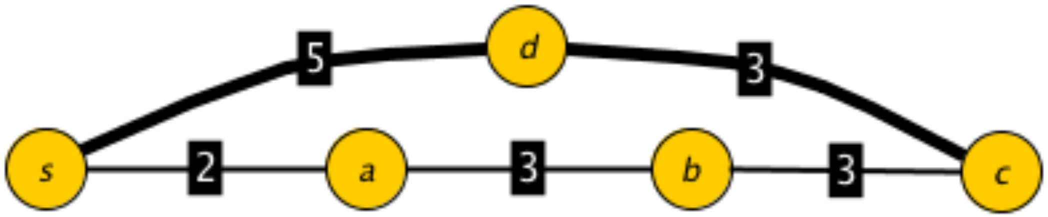


- Dare lo pseudo-codice di un algoritmo efficiente che data una rete e due nodi A e B, trova un cammino tra A e B con il minimo livello d'interferenza. Discutere la correttezza e la complessità dell'algoritmo proposto. *Suggerimento: modificare opportunamente l'algoritmo di Dijkstra.*

ESERCIZIO 5

Un cammino da un nodo u a un nodo v si dice super-minimo se ha peso minimo tra tutti i cammini da u a v e inoltre tra tutti i cammini di peso minimo da u a v ha il minimo numero di archi. Dato un grafo pesato G tale che i pesi sono interi positivi, si vogliono trovare i cammini super-minimi da un nodo s.

Ad esempio, nel grafo qui sotto vogliamo il cammino (s, d, c) e non quello di pari peso ma più lungo (s, a, b, c).



Mostrare come modificare i pesi del grafo G in modo tale che applicando Dijkstra al grafo coi nuovi pesi si ottengono i cammini super minimi di G.

ESERCIZIO 6

In un grafo con nodi colorati diciamo che un arco è felice se i suoi estremi hanno differenti colori. Disponendo dei soli colori rosso e nero, si desidera colorare i nodi di un grafo G in modo da massimizzare il numero di archi felici. Per risolvere il problema viene proposto il seguente algoritmo greedy:

```
INPUT un grafo non diretto G
    FOR ogni nodo u di G DO
        rossi ← il numero di nodi colorati di rosso adiacenti a u
        neri ← il numero di nodi colorati di nero adiacenti a u
        IF (rossi < neri) THEN
            colora di rosso il nodo u
        ELSE
            colora di nero il nodo u
    ENDFOR
```

- Descrivere un'implementazione dell'algoritmo che abbia complessità $O(n + m)$.
- Provare che l'algoritmo è corretto o fornire un controesempio.

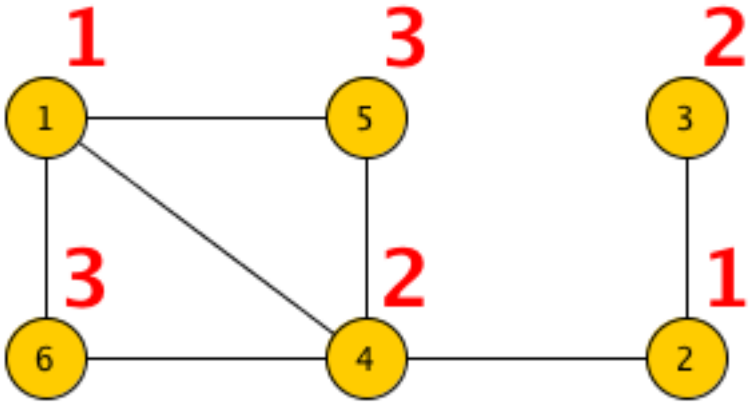
ESERCIZIO 7

Dato un grafo non diretto $G = (\{1, 2, \ldots n\}, E)$ vogliamo assegnare ad ogni nodo un valore in $\{1, 2, \ldots, n\}$ in modo tale che nodi adiacenti ricevono valori distinti e che il numero di valori distinti usati sia minimo.
Viene proposto il seguente algoritmo greedy che restituisce il vettore SOL dove $SOL[i]$ rappresenta il valore assegnato al nodo i:

```
FOR i = 1 TO n DO
        C ← {1, 2, ..., n}
        FOR j = 1 TO i − 1 DO
                IF ({i, j} ∈ E) THEN C ← C − {SOL[j]}
        ENDFOR
        SOL[i] = 1
        WHILE SOL[i] ∉ C DO SOL[i] ← SOL[i] + 1
ENDFOR
OUTPUT SOL
```

Dire se l'algoritmo proposto risolve il problema. In caso affermativo fornire una prova di correttezza altrimenti fornire un controesempio.

Ad esempio ai 6 nodi del grafo in basso l'algoritmo di colorazione greedy proposto assegna i valori evidenziati in rosso utilizzando così solo 3 valori distinti.



In questo caso la soluzione prodotta è ottima infatti nel grafo sono presenti cicli da 3 (ad esempio in nodi 1, 4 e 5) e quindi per colorare i tre nodi sono necessari almeno 3 colori.
In generale se nel grafo da colorare è presente un sottografo completo di k nodi allora per colorare il grafo servono almeno k colori.

ESERCIZIO 8

Un *accoppiamento* in un grafo è un sottoinsieme dei suoi archi tale che nessuna coppia di archi del sottoinsieme condivide uno stesso nodo. Dato un grafo G vogliamo trovare un accoppiamento di massima cardinali. Viene proposto il seguente algoritmo greedy

$SOL \leftarrow \emptyset$
WHILE $E \neq \emptyset$ DO
  in E prendi arco $\{x, y\}$ con x di grado minimo
  $SOL \leftarrow SOL \cup \{\{x, y\}\}$
  cancella da Egli archi che hanno per estremo x o y
ENDWHILE
OUTPUT SOL

- Mostrare che l'algoritmo non produce necessariamente un accoppiamento di massima cardinalità per il grafo
- Mostrare che nel caso di grafi aciclici l'algoritmo è corretto.

ESERCIZIO 9

Dato un grafo non diretto G, chiamiamo *triangolo* un qualsiasi insieme di tre nodi distinti mutuamente adiacenti. Si desidera trovare un triangolo di peso minimo (il peso di un triangolo è dato dalla somma dei pesi dei tre archi che lo compongono) in un grafo pesato e completo. Per risolvere il problema viene proposto il seguente algoritmo greedy:

INPUT un grafo $G = (V, E)$ non diretto, pesato e completo
Seleziona un arco $\{u, v\}$ di G di peso minimo
Seleziona un arco $\{x, y\}$ di peso minimo tra quelli incidenti a $\{u, v\}$
RETURN il triangolo determinato dagli archi $\{u, v\}$ e $\{x, y\}$

- Descrivere un'implementazione dell'algoritmo che abbia complessità $O(n^2)$.
- Provare che l'algoritmo è corretto o fornire un controesempio.

ESERCIZIO 10

Un farmacista ha P pillole e dispone di n flaconi di differenti costi e capacità. L'i-esimo flacone è caratterizzato dalla coppia di interi (c_i, d_i) dove c_i e d_i sono il costo e la capacità del flacone i, rispettivamente. Per ognuno dei seguenti due problemi dimostrare o confutare (tramite un controesempio) che la strategia proposta risolve il problema.

1. *Per sistemare le P pillole usando il minor numero di flaconi*, si propone la seguente strategia greedy:

   > Finché hai pillole da sistemare, seleziona tra i flaconi rimasti quello di massima capacità e inserisci in questo flacone il numero massimo di pillole.

2. *Per sistemare le P pillole minimizzando il costo dei flaconi usati*, si propone la seguente strategia greedy:

   > Finché hai pillole da sistemare, seleziona tra i flaconi rimasti quello di costo minimo e inserisci in questo flacone il numero massimo di pillole.

ESERCIZIO 11

Sia $G = (V, E)$ un grafo non orientato connesso e con pesi sugli archi. Sia T un minimo albero di copertura per G.

a. Provare che T è ancora un minimo albero di copertura anche per il grafo che si ottiene da G incrementando di una stessa costante c il peso degli archi.

b. Provare che T deve contenere un arco di peso minimo.

c. Provare che qualora i pesi di G siano tutti distinti, allora T è l'unico minimo albero di copertura. Mostrare che che la condizione sul fatto che i pesi siano tutti distinti non è necessaria per l'unicità del minimo albero di copertura.

ESERCIZIO 12

- Sia G un grafo non diretto, connesso e pesato. Dimostrare o confutare che un albero dei cammini minimi di G è anche un albero di copertura di peso minimo.

- Sia G un grafo non diretto, connesso e pesato dove i pesi sono tutti diversi tra loro. Sia T un minimo albero di copertura di G, sia s un nodo e T_s l'albero dei cammini minimi da s verso tutti gli altri nodi di G. Dimostrare oppure fornire un controesempio che T_s e T condividono almeno un arco.

ESERCIZIO 13

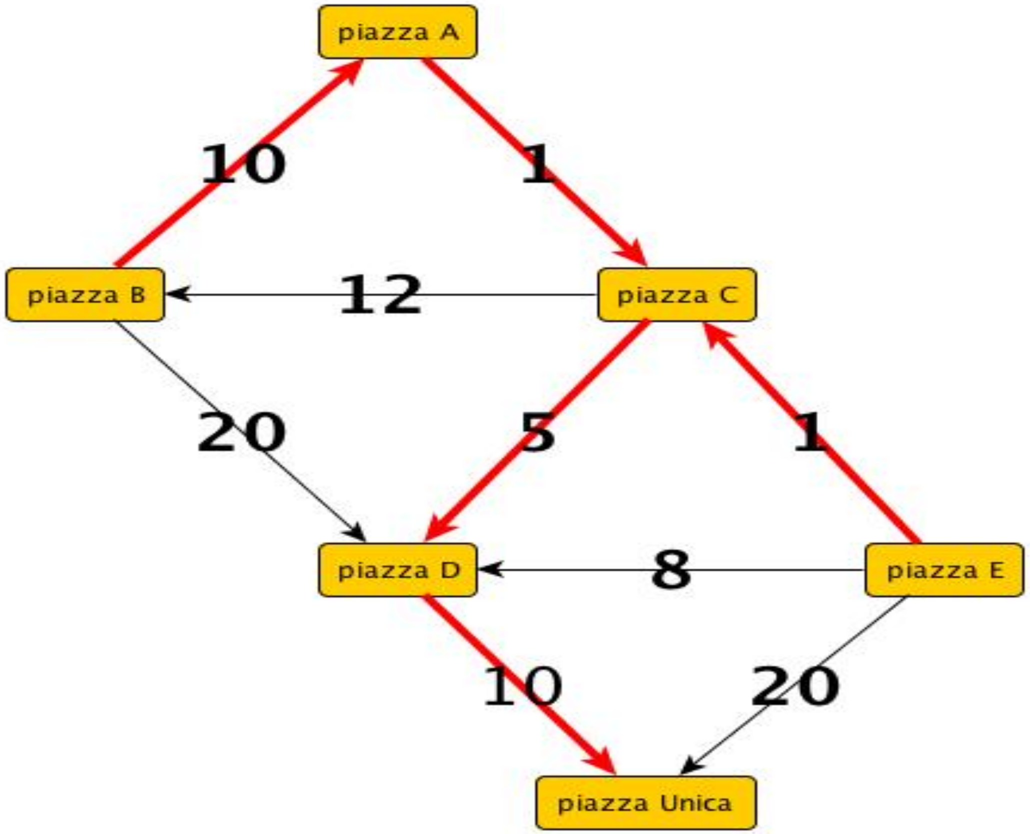
La rete viaria della città Universa consiste di varie piazze collegate fra loro da strade. La particolarità di Universa è che tutte le strade sono a senso unico. La piazza *Unica* è la piazza più importante e si sa che essa è raggiungibile da qualsiasi altra piazza della città.
Il sindaco vuole selezionare un insieme di strade la cui lunghezza totale sia minima e che permettano da una qualsiasi piazza di raggiungere la piazza Unica. Gli urbanisti pensano di trovare questo insieme di strade tramite un algoritmo di Prim modificato che partendo dalla piazza Unica ad ogni passo sceglie sempre la strada entrante di lunghezza minima.

- L'algoritmo degli urbanisti produce sempre un insieme di strade che garantisce la raggiungibilità della piazza Unica da una qualsiasi altra piazza?

- L'insieme selezionato ha lunghezza totale minima?

Motivare bene le risposte.

- nel grafo in figura sono evidenziate in rosso le strade selezionate dall'algoritmo di Prim modificato.
- la soluzione prodotta ha lunghezza totale 27 e garantisce che da qualunque piazza si possa ancora raggiungere la piazza Unica.



ESERCIZIO 14

Ci sono n case, indicate con gli interi $i = 1, 2, \dots, n$, ognuna delle quali necessita di una fornitura d'acqua. La costruzione di un pozzo nella casa i costa $p[i]$ e la costruzione di una tubazione fra le case i e j costa $c[i, j]$. La fornitura d'acqua per una casa o è un pozzo costruito nella casa o è un cammino di tubazioni dalla casa a qualche pozzo.

Dare lo pseudo-codice di un algoritmo che determina le case in cui costruire i pozzi e le tubazioni da costruire per dare una fornitura d'acqua a tutte le case minimizzando il costo totale.

L'algoritmo deve avere complessità $O(n^2)$. Discutere la correttezza dell'algoritmo.

Suggerimento: rappresentare il problema tramite un opportuno grafo pesato con $n+1$ *nodi, il nodo in più rappresenta i pozzi ...*