Algoritmi

Esercizi di esame

Marcella Anselmo

a partire da Gennaio 2010

1 Appello 13 Gennaio 2010

1. Utilizzando lo spazio a disposizione, spiegare in maniera precisa, cosa e' un algoritmo, a cosa serve, e quali sono gli aspetti principali di un algoritmo ai quali ci siamo interessati nell'ambito del corso.

2. Si considerino le seguenti funzioni: $\sqrt{n^3} + 2\log n$, $\log n + \log\log n$, 3^n, $n^{\log^2 n}$, $(2^{\log_2 n})^5$, n^{15}.

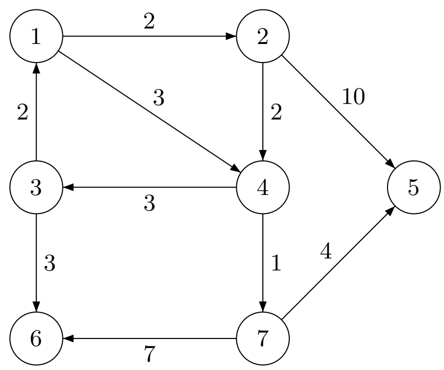
   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   b) Si dimostri formalmente (cioe' fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i.

3. Dati un intero k e un array $A[1..n]$ di interi positivi ordinati **in ordine crescente**, descrivere ed analizzare un algoritmo per trovare una coppia di posizioni distinte i, j tali che $1 \leq i < j \leq n$ e $A[i] \times A[j] = k$. Il tempo di esecuzione dell'algoritmo deve essere $O(n)$ nel caso pessimo.

   E' necessario spiegare verbalmente il funzionamento dell'algoritmo proposto, indicare la tecnica di progettazione usata e giustificare la correttezza dell'algoritmo.

4. Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l'algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.



5. Descrivere ed analizzare un algoritmo che, preso in input un grafo, determini se esso contiene o meno un ciclo di lunghezza pari, ed in caso affermativo ne fornisca uno.

   E' necessario giustificare la correttezza dell'algoritmo, citando i risultati teorici su cui si basa.

2 Appello 26 Gennaio 2010

1. Utilizzando lo spazio a disposizione, descrivere le principali tecniche di progettazione di algoritmi studiate, confrontandone quelli che secondo voi sono i punti salienti.

2. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 5n^2 + 11\log n$ e $T_{A'}(n) = \sqrt{(n^3)}logn + n$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

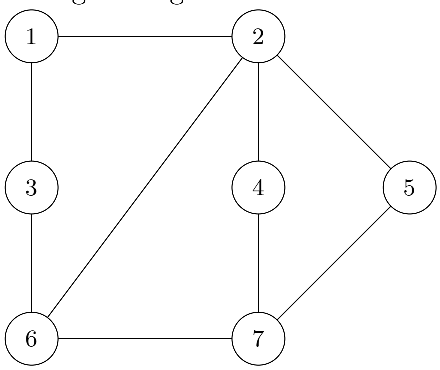
3. In una (ignota) localita' sciistica, vi sono n stazioni s_1, s_2, $\cdots$, s_n, collegate tra loro da piste da sci. Dalla stazione s_1, in vetta alla montagna, e' possibile arrivare alla stazione s_n a valle in vari modi. Poiche' le stazioni s_1, s_2, $\cdots$, s_n, sono ordinate secondo la loro altezza decrescente, una pista che parte da s_i arriva in una stazione s_j, con $j > i$; inoltre per ogni $1 \le i < j \le n$ c'e' una pista da s_i a s_j a cui e' associato un tempo di percorrenza $t_{i,j}$. Poiche' le piste hanno diverso grado di difficolta', e' possibile che per andare da s_i a s_j, si impieghi meno tempo utilizzando piu' piste collegate che non quella diretta.

   Si consideri il problema di determinare il tempo minimo per andare da s_1 ad s_n, avendo in input i tempi di percorrenza $t_{i,j}$,

   a) Indicare qual e' il tempo di esecuzione di un algoritmo di "forza bruta" che risolve il problema.

   b) Descrivere un algoritmo di programmazione dinamica per lo stesso problema. Analizzare la complessita' dell'algoritmo proposto e confrontarla con quella del punto a).

4. Sia G il grafo seguente:



   a) Eseguire la BFS su G a partire dal nodo 1, mostrando i "layers" (strati) ottenuti e l'albero BFS risultante.

   b) Dire se G e' un grafo bipartito, motivando la risposta e, in caso affermativo, fornire la partizione dei vertici per cui G e' un grafo bipartito.

5. Fornire l'esempio di un grafo e di un suo vertice, in modo che l'albero BFS abbia altezza 1, mentre l'albero DFS abbia altezza pari al numero di vertici meno 1. Giustificare la risposta.

3 Appello 19 Febbraio 2010

1. Utilizzando lo spazio a disposizione, spiegare cosa si intende per tempo di esecuzione di un algoritmo, in cosa consiste la sua analisi asintotica e quali strumenti matematici necessita.

2. a) Si definisca cosa si intende per codice ottimale per un insieme di caratteri cui sono assegnate delle frequenze.

   b) Si determini il codice ottimale per l'insieme di caratteri $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 15$, $f[b] = 30$, $f[c] = 7$, $f[d] = 20$, $f[e] = 24$, $f[g] = 4$, fornito dall'algoritmo di Huffman studiato. E' sufficiente disegnare l'albero risultante e indicare la stringa associata ad ogni carattere di C.

3. Si supponga di dover effettuare un viaggio dalla localita' A alla localita' B con una moto con un'autonomia di 100 chilometri a serbatoio pieno. Lungo il percorso da A a B si incontrano n distributori di carburante. Sia d_0 la distanza da A al primo distributore, d_i la distanza che

separa il distributore i dal distributore $i+1$, per $i = 1, 2, \cdots, n-1$ e d_n la distanza dell'ultimo distributore da B; inoltre d_i e' inferiore a 100 chilometri, per ogni $i = 0, 1, \cdots, n$.

a) Quale puo' essere un criterio di scelta per minimizzare il numero di distributori in cui fermarsi? (*Suggerimento: in quale distributore conviene fermarsi la prima volta?*)

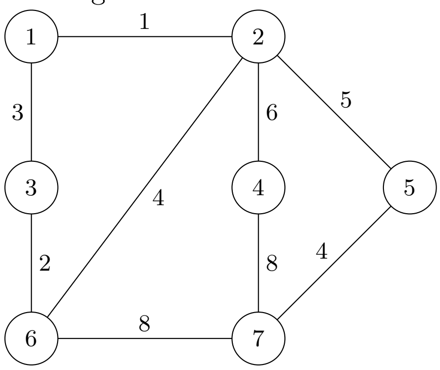
b) Descrivere un algoritmo greedy basato sul criterio proposto.

c) Giustificare la correttezza dell'algoritmo proposto.

4. Sia G il grafo seguente.

   a) Eseguire l'algoritmo di Kruskal su G

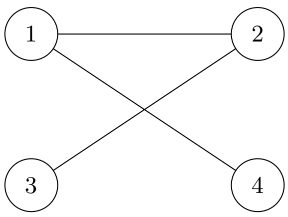
   b) Trovare un k-clustering per G con $k = 3$, dopo avere completato G con gli archi mancanti cui si assegni costo 100.



5. a) Definire cos'e' un matching massimale in un grafo bipartito.

   b) Fornire un esempio di un grafo bipartito con due distinti matching massimali.

   c) Trovare un matching massimale per il seguente grafo $G = (V, E)$ bipartito, eseguendo l'algoritmo studiato.



4 Appello 12 Aprile 2010

1. Utilizzando lo spazio a disposizione, spiegare in maniera precisa, cosa e' una relazione di ricorrenza, a cosa serve, con quali metodi e' possibile risolverla. Fornire infine un esempio, a scelta, di una relazione di ricorrenza e della sua risoluzione.

2. Si considerino le seguenti funzioni: $\sqrt{n^3} + 2\log n$, $\log n + \log\log n$, 3^n, $n^{\log n}$, 2^n, n^{15}.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

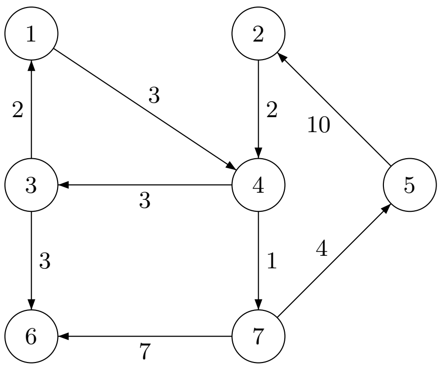
   b) Si dimostri formalmente (cioe' fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i.

3. Si consideri un algoritmo *greedy* (goloso) per il problema della selezione delle attivita' basato sulla scelta dell'attivita' che inizia per ultima.

   a) Se pensate che tale algoritmo restituisca sempre una soluzione ottimale, dimostratelo formalmente.

   a') Se pensate che tale algoritmo non sempre restituisca una soluzione ottimale, mostrate un contro-esempio.

4. Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l'algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.



5. Descrivere ed analizzare un algoritmo che, preso in input un grafo, determini se esso contiene o meno un ciclo di lunghezza pari, ed in caso affermativo ne fornisca uno.

   E' necessario giustificare la correttezza dell'algoritmo, citando i risultati teorici su cui si basa.

5 Appello 16 Giugno 2010

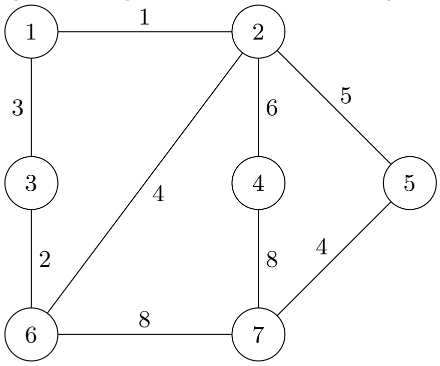
1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 2T(n/2) + n^2$ e $T_{A'}(n) = T_{A'}(n-1) + n$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. Sia dato un vettore di n elementi che possono assumere solo i tre valori VERDE, BIANCO e ROSSO. Descrivere ed analizzare un algoritmo per ordinare il vettore in tempo $\Theta(n)$ in modo che tutti gli elementi con valore VERDE precedano tutti quelli con valore BIANCO, che a loro volta precedono tutti quelli con valore ROSSO, tenendo conto che le uniche operazioni ammesse sono l'esame di un elemento e lo scambio di due elementi, dati i loro indici.

3. a) Descrivere brevemente il problema dello zaino.

b) Descrivere un algoritmo *greedy* che risolva il problema dello zaino, nelle ipotesi che tutti gli oggetti abbiano lo stesso peso. Dimostrare la correttezza dell'algoritmo proposto.

4. Eseguire l'algoritmo di Prim sul grafo G seguente.



5. La *larghezza* di un albero (radicato) e' il numero massimo di nodi che si trovano allo stesso livello (o profondita'). Si modifichi l'algoritmo di visita BFS per calcolare la larghezza di un albero.

6 Appello 5 Luglio 2010

1. Dimostrare formalmente le seguenti affermazioni, esibendo le costanti opportune.
   (a) $p_1(n) = 5n - 6 = \Theta(n)$
   (b) $p_2(n) = 2n^2 - 5n + 6 = \Theta(n^2)$
   (c) $p_3(n) = n^3 + 2n^2 - 5n + 6 = \Theta(n^3)$.

2. Sia $V[1..n]$ un vettore ordinato di 0 e 1. Descrivere ed analizzare un algoritmo per determinare il numero di 0 presenti in $V[1..n]$ in tempo $O(\log n)$.

3. Si consideri una strada dritta di lunghezza k, punteggiata da n abitazioni collocate ai chilometri $k_1, k_2, \cdots, k_n$, con $0 \leq k_i \leq k$, per ogni $i = 1, 2, \cdots, n$. Una compagnia telefonica vuole collocare un insieme di torri cellulari in modo tale che ogni abitazione sia ad una distanza massima d da una torre.

   a) Descrivere ed analizzare un algoritmo per risolvere il problema utilizzando il numero minimo di torri.

   b) Giustificarne la correttezza.

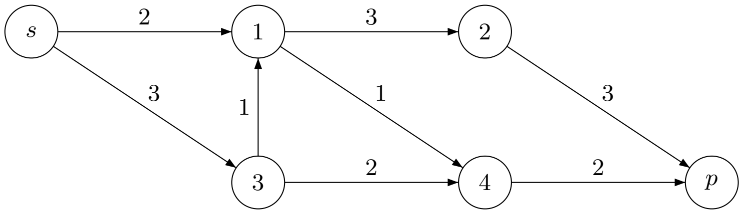
4. Disegnare un grafo con 6 nodi per il quale l'albero BFS sia diverso dall'albero DFS.

5. a) Definire cos'e' una rete di flusso.

   b) Definire cos'e' un taglio minimo per una rete di flusso.

   c) Elencare tutti i tagli minimi nella seguente rete di flusso.

   d) Spiegare come e' possibile trovare un taglio minimo in una rete di flusso utilizzando l'algoritmo di Ford-Fulkerson.



7 Appello 7 Settembre 2010

1. Risolvere la seguente relazione di ricorrenza:

   $T(1) = 5$

   $T(n) = 2T(n-1) + 3$ se $n \geq 2$.

2. Sia $A[1, ..., n]$ un vettore ordinato che e' stato shiftato k posizioni a destra. Ad esempio, il vettore $[35, 42, 1, 7, 15, 18, 28, 30]$ e' un vettore ordinato che e' stato shiftato $k = 2$ posizioni a destra, mentre il vettore $[15, 18, 28, 30, 35, 42, 1, 7]$ e' un vettore ordinato che e' stato shiftato $k = 6$ posizioni a destra.

   (a) Supponendo di avere A e k in input, descrivere un algoritmo che determini il massimo in A in tempo $O(1)$.

   (b) Supponendo di avere solo il vettore A in input, descrivere un algoritmo che determini il massimo in A in tempo $O(\log n)$.

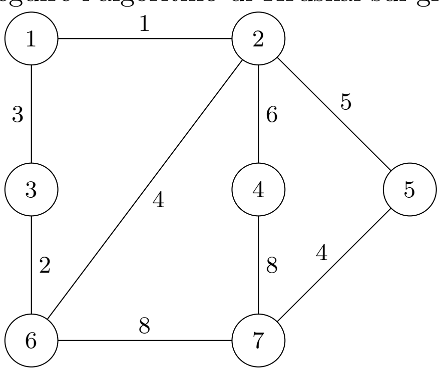
3. Si consideri una strada dritta di lunghezza k, punteggiata da n abitazioni collocate ai chilometri $k_1, k_2, \cdots, k_n$, con $0 \leq k_i \leq k$, per ogni $i = 1, 2, \cdots, n$. Una compagnia telefonica vuole collocare un insieme di torri cellulari in modo tale che ogni abitazione sia ad una distanza massima d da una torre.

   a) Descrivere ed analizzare un algoritmo per risolvere il problema utilizzando il numero minimo di torri.

   b) Giustificarne la correttezza.

4. Descrivere un algoritmo per determinare se un grafo diretto $G = (V, E)$ è fortemente connesso, argomentare la sua correttezza e valutarne la complessità di tempo.

5. Eseguire l'algoritmo di Kruskal sul grafo seguente.



8 Appello 12 Novembre 2010

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 2T(n/2) + 3$ e $T_{A'}(n) = T_{A'}(n-1) + 5$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. a) Descrivere gli aspetti essenziali della tecnica Divide et Impera, utilizzando lo spazio designato.

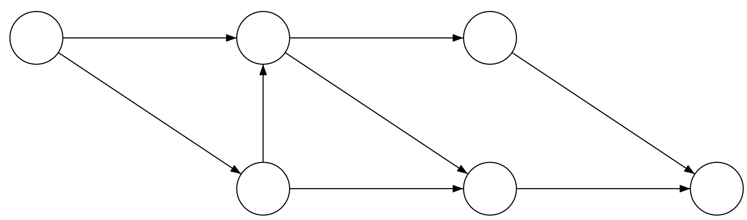
   b) Descrivere ed analizzare un algoritmo **basato sulla tecnica Divide et Impera** che dato un array $A[1, ..., n]$ di interi ne restituisca il massimo.

3. Una societa' elettrica vuole piazzare dei lampioni lungo un viale rettilineo dove sono situate delle villette, in modo da illuminare tutte le villette. Ogni lampione riesce ad illuminare tutta la zona compresa nel raggio di 50 metri. Siano $\{v_1, v_2, \cdots, v_n\}$ le villette e per ogni $i = 1, \cdots, n-1$, $d_{i,i+1}$ la distanza della viletta v_i da v_{i+1}.

   a) Descrivere ed analizzare un algoritmo che presi in input i valori $d_{i,i+1}$ restituisca il numero minimo di lampioni necessari per illuminare tutte le villette.

   b) Argomentare la correttezza dell'algoritmo.

4. Fornire un ordinamento topologico per il seguente DAG (*Directed Acyclic Graph*), utilizzando l'algoritmo studiato e spiegando le operazioni via via eseguite.



5. a) Definire il problema del cammino minimo (*Shortest Path Problem*) in un grafo.

   b) L'algoritmo di Bellman-Ford e' un algoritmo di programmazione dinamica per il calcolo del cammino minimo in un grafo. Qual e' la sua complessita' di tempo?

   c) Descrivere un algoritmo di programmazione dinamica per il calcolo del cammino minimo in un DAG (*Directed Acyclic Graph*) che sia piu' efficiente dell'algoritmo di Bellman-Ford.

9 Appello 20 Gennaio 2011

1. Si risolva la seguente relazione di ricorrenza:

   $T(n) = 2T(n/3) + cn$ con $T(1) = b$.

2. a) Si descrivano (soltanto) l'input e l'output della procedura PARTITION studiata.

   b) Descrivere un algoritmo che, dati un array $A[1..n]$ di interi positivi distinti e un intero k con $1 \leq k \leq n$, calcoli il k-esimo minimo di A. Tale algoritmo non dovra' fare alcun ricorso ad algoritmi di ordinamento, ma potra' utilizzare la procedura PARTITION.

   E' necessario spiegare verbalmente il funzionamento dell'algoritmo proposto, indicare la tecnica di progettazione usata e giustificare la correttezza dell'algoritmo.

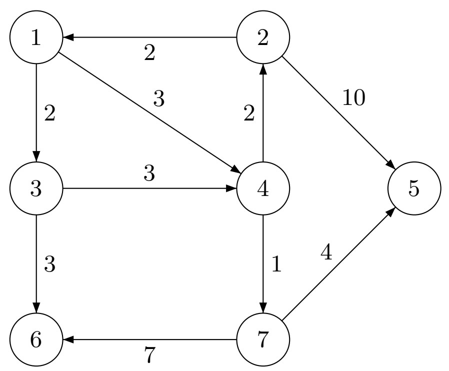
   Esempio: se $A[1..7] = [6, 9, 3, 2, 5, 7, 8]$ e $k = 3$ l'algoritmo dovra' restituire 5.

   c) Analizzare il caso peggiore dell'algoritmo proposto al punto b).

3. a) Si descrivano (soltanto) l'input e l'output dell'algoritmo Huffman studiato.

   b) Si determini il codice ottimale per l'insieme di caratteri $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 15$, $f[b] = 30$, $f[c] = 7$, $f[d] = 20$, $f[e] = 24$, $f[g] = 4$, fornito dall'algoritmo Huffman studiato. E' sufficiente disegnare l'albero risultante e indicare la stringa associata ad ogni carattere di C.

4. Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l'algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.



5. Il lungomare di Salermo e' un viale a senso unico che congiunge la stazione con il porto sul quale vi sono n fermate di autobus $f_1, f_2, \cdots, f_n$. Il percorso e' servito da vari autobus, ognuno dei quali fa alcune fermate intermedie, ma non necessariamente tutte. Si supponga che il tempo di percorrenza per andare dalla fermata f_i alla fermata f_j con $i > j$ sia costante (perche' dipende dalla lunghezza del tratto e dal traffico stimato) ed uguale a $t(i, j)$. E' possibile che vi siano dei tratti fra due fermate non coperti da autobus.

   Si consideri il problema di determinare il tempo minimo per andare con uno o piu' autobus dalla stazione al porto di Salermo, avendo in input i tempi di percorrenza $t(i, j)$ per ogni $1 \leq i < j \leq n$ tale che il tratto da f_i a f_j sia coperto da un autobus.

   a) Indicare qual e' il tempo di esecuzione nel caso peggiore di un algoritmo di "forza bruta" che risolve il problema.

   b) Descrivere un algoritmo di programmazione dinamica per lo stesso problema. Analizzare la complessita' dell'algoritmo proposto e confrontarla con quella del punto a).

10 Appello 9 Febbraio 2011

1. Si considerino le seguenti funzioni: $\sqrt{\log n} + \log n$, $\log \log n$, 2^n, $4^{\log_2 n}$, $n^2 - 3n + 7$, $n + 15$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   b) Si dimostri formalmente (cioe' fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i.

2. a) Fornire lo pseudocodice di un algoritmo ricorsivo che ordini un array $A[1..n]$ nel seguente modo: prima ordina ricorsivamente $A[1..n-1]$ e poi inserisce $A[n]$ nell'array ordinato $A[1..n-1]$.

   b) Analizzare la complessita' di tempo dell'algoritmo proposto al punto b).

3. I numeri di Tribonacci sono cosi' definiti:

   $R(0) = 0$

   $R(1) = 0$

   $R(2) = 1$

   $R(n) = R(n-1) + R(n-2) + R(n-3)$ se $n \geq 3$.

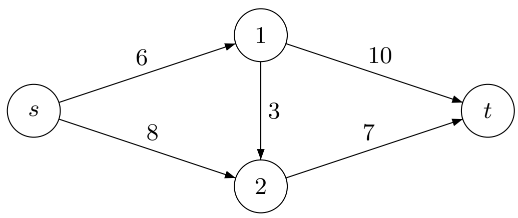
   a) Scrivere lo pseudocodice di un algoritmo di programmazione dinamica per il calcolo dell'n-esimo numero di Tribonacci $R(n)$.

   b) Analizzare la complessita' di tempo e di spazio dell'algoritmo proposto.

   c) E' possibile realizzare l'algoritmo con spazio $O(1)$? Giustificare la risposta.

4. Descrivere ed analizzare un algoritmo che, dati un grafo $G = (V, E)$, una funzione costo che assegna un intero positivo $c(e)$ ad ogni arco $e \in E$, e due nodi $u, v \in V$, calcoli i costi dei cammini minimi da u, che non passino da v.

5. Si esegua l'algoritmo di Ford- Fulkerson sulla seguente rete di flusso, indicando gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto ed un taglio di capacita' minima.



11 Appello 25 Febbraio 2011

1. Si consideri il seguente algoritmo:

```
ALGORITMO(n)
if n = 1 then return 0
         else ALGORITMO(n/2)
                x <-- 0
                for i=1,..,n do
                        x <--x + 2i
```

   Si esprima la complessita' dell'algoritmo mediante una relazione di ricorrenza e se ne dia una soluzione.

2. Sia dato un vettore binario ordinato $A[1..n]$.

   (a) Progettare un algoritmo di complessita' $\Theta(n)$ nel caso peggiore, che conti il numero di occorrenze di 1 nel vettore A.

   (b) Progettare un algoritmo di complessita' $O(\log n)$, che conti il numero di occorrenze di 1 nel vettore A.

3. Un allenatore di vela deve scegliere la squadra per la prossima importante regata. L'allenatore puo' scegliere fra n atleti a_1, a_2, ... a_n. Ad ogni atleta a_i ha assegnato un voto v_i secondo la sua bravura, ma deve tenere conto pure che la barca a vela puo' sostenere un peso complessivo che non superi P ed ogni atleta ha un suo peso p_i.

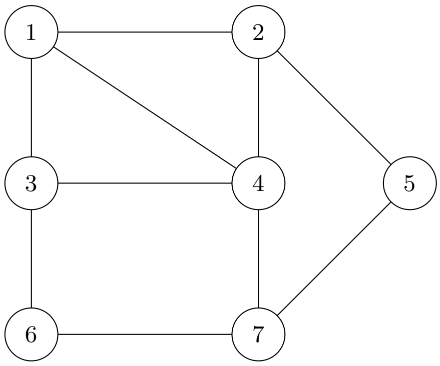
   Descrivere ed analizzare un algoritmo per determinare la squadra migliore (cioe' con massima bravura totale) che possa salire sulla barca (senza superare il limite di peso). Specificare la tecnica di programmazione usata.

   Suggerimento: notate analogie con qualche problema studiato?

4. Sia $G = (V, E)$ il seguente grafo.

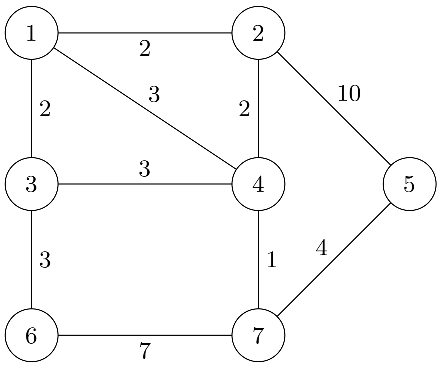
   a) Fornire una rappresentazione di G con le matrici di incidenza.

   b) Fornire una rappresentazione di G con le liste di adiacenza.



5. a) Eseguire l'algoritmo di Prim sul seguente grafo $G = (V, E)$ partendo dal nodo $s = 1$, mostrando gli aggiornamenti effettuati.

   b) Eseguendo l'algoritmo a partire da un altro nodo, avremmo ottenuto un risultato finale diverso?



12 Appello 14 Giugno 2011

1. Si considerino le seguenti funzioni: $\sqrt{\log n^3}$, $\log \log n$, 3^n, $4^{\log_2 n}$, $n^3 + 9$, $3n - 1$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

b) Si dimostri formalmente (cioe' fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i.

2. Sia $A[1..n]$ un vettore ordinato che e' stato shiftato k posizioni a destra. Ad esempio, il vettore $[35, 42, 1, 7, 15, 18, 28, 30]$ e' un vettore ordinato che e' stato shiftato $k = 2$ posizioni a destra, mentre il vettore $[15, 18, 28, 30, 35, 42, 1, 7]$ e' un vettore ordinato che e' stato shiftato $k = 6$ posizioni a destra.

   (a) Supponendo di avere A e k in input, dare un algoritmo che determina il massimo in A in tempo $O(1)$.

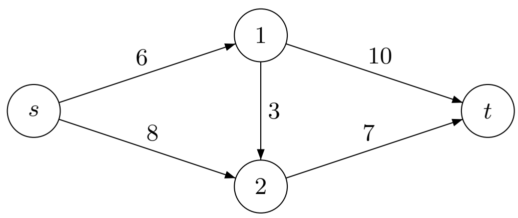
   (b) Supponendo di avere solo il vettore A in input, dare un algoritmo che determina il massimo in A in tempo $O(logn)$.

3. a) Si descrivano (soltanto) l'input e l'output dell'algoritmo Huffman studiato.

   b) Si determini il codice ottimale per l'insieme di caratteri $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 15$, $f[b] = 27$, $f[c] = 8$, $f[d] = 20$, $f[e] = 26$, $f[g] = 4$, fornito dall'algoritmo Huffman studiato. E' sufficiente disegnare l'albero risultante e indicare la stringa associata ad ogni carattere di C.

4. Descrivere ed analizzare un algoritmo che, dati un grafo $G = (V, E)$, e un nodo $u \in V$, restituisca tutti i vertici a distanza massima da u.

5. Si esegua l'algoritmo di Ford- Fulkerson sulla seguente rete di flusso, indicando gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto ed un taglio di capacita' minima.



13 Appello 4 Luglio 2011

1. Si risolva la seguente relazione di ricorrenza.

   $T(n) = T(n-1) + n/2$ e $T(1) = 1$ nelle ipotesi che n sia una potenza di 2.

2. Sia dato un vettore ordinato $A[1..n]$ di interi distinti. Progettare un algoritmo che determini, in tempo $O(\log n)$, se esiste o meno un intero i tale che $A[i] = i$.

3. I numeri di Tribonacci sono cosi' definiti:

   $R(0) = 0$

   $R(1) = 0$

   $R(2) = 1$

   $R(n) = R(n-1) + R(n-2) + R(n-3)$ se $n \geq 3$.

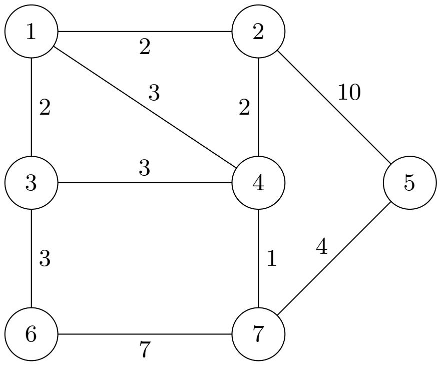
   a) Scrivere lo pseudocodice di un algoritmo di programmazione dinamica per il calcolo dell'n-esimo numero di Tribonacci $R(n)$.

b) Analizzare la complessita' di tempo e di spazio dell'algoritmo proposto.

c) E' possibile realizzare l'algoritmo con spazio $O(1)$? Giustificare la risposta.

4. a) Descrivere il problema risolto dall'algoritmo di Dijkstra.

   b) Eseguire l'algoritmo di Dijkstra sul seguente grafo $G = (V, E)$ partendo dal nodo $s = 1$, mostrando gli aggiornamenti effettuati.



5. Descrivere un algoritmo che dato un grafo $G = (V, E)$, un vertice $v \in V$ e un intero k, restituisca il numero di vertici di V a distanza minore o uguale a k.

14 Appello 9 Settembre 2011

1. Si risolva la seguente relazione di ricorrenza:

   $T(n) = 2T(n/2) + n$ con $T(1) = 1$.

2. Si consideri la procedura PARTITION che, preso in input un array $A[1..n]$ con $x = A[n]$, permuta l'array in modo che, al termine dell'esecuzione restituisca un indice q tale che, in posizione q vi sia x, mentre ogni elemento a sinistra della posizione q contenga elementi minori o uguali ad x e ogni elemento a destra di q contenga elementi maggiori o uguali ad x.

   Descrivere un algoritmo ricorsivo che, dati un array $A[1..n]$ di interi positivi distinti e un intero k con $1 \leq k \leq n$, calcoli il k-esimo minimo di A.

   Esempio: Se $A[1..7] = [6, 9, 3, 2, 5, 7, 8]$ e $k = 3$ l'algoritmo dovra' restituire 5.

   Tale algoritmo non dovra' fare alcun ricorso ad algoritmi di ordinamento, ma dovra' basarsi sulla procedura PARTITION.

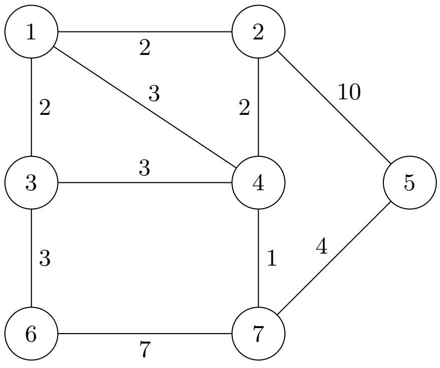
   E' necessario inoltre: spiegare verbalmente il funzionamento dell'algoritmo proposto, indicare la tecnica di progettazione usata, giustificare la correttezza dell'algoritmo ed analizzare il caso peggiore dell'algoritmo proposto.

3. a) Si descrivano (soltanto) l'input e l'output dell'algoritmo HUFFMAN studiato.

   b) Si determini il codice ottimale per l'insieme di caratteri $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 14$, $f[b] = 30$, $f[c] = 8$, $f[d] = 20$, $f[e] = 24$, $f[g] = 4$, fornito dall'algoritmo HUFFMAN studiato. E' sufficiente disegnare l'albero risultante e indicare la stringa associata ad ogni carattere di C.

4. a) Spiegare cosa calcola l'algoritmo di Kruskal richiamato su un grafo pesato e quali sono le principali differenze con l'algoritmo di Prim.

   b) Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l'algoritmo di Kruskal su G, mostrando gli aggiornamenti via via effettuati.



5. Descrivere ed analizzare un algoritmo efficiente che, dato un grafo (non orientato) $G = (V, E)$, restituisca il numero di componenti connesse di G.

15 Appello 5 Novembre 2011

1. Senza far ricorso al Master Theorem, si risolva la seguente relazione di ricorrenza, supponendo che n sia una potenza di 2 e che $T(1) = 1$:

$$T(n) = 4T(n/2) + 3n$$

2. Descrivere ed analizzare un algoritmo basato sul paradigma *divide et impera* che dato un vettore *ordinato* $A[1..n]$ di interi strettamente positivi (cioe' per ogni $1 \leq i \leq n$, $A[i] \geq 1$), restituisca il numero di occorrenze di 1 nel vettore A. L'algoritmo deve avere complessita' di tempo $O(\log n)$.

3. a) Definire il problema della ricerca dei cammini minimi in un grafo.

   b) Descrivere l'algoritmo di Bellman e Ford per la soluzione del problema di cui al punto a).

4. Si consideri un grafo pesato con costi (a due a due) distinti sugli archi.

   a) Enunciare la proprieta' del taglio.

   b) Dimostrare che l'arco di peso minimo del grafo, appartiene a qualsiasi minimum spanning tree del grafo.

5. a) Definire cos'e' un grafo bipartito.

   b) Definire cos'e' un matching massimale in un grafo bipartito.

   Si consideri il seguente grafo G.

   c) Dimostrare che G e' bipartito, eseguendo l'algoritmo di test studiato.

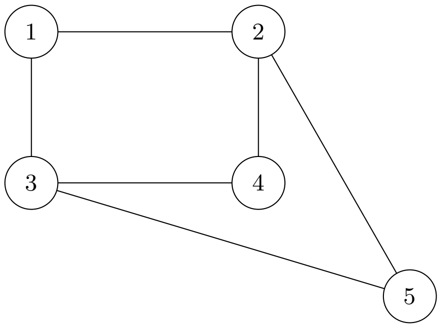
   d) Determinare un matching massimale di G, eseguendo l'algoritmo studiato.

16 Appello 13 Gennaio 2012

1. Si considerino le seguenti funzioni: $\log n - 1$, 2^n, $3^{\log_2 n}$, $n^2 + 3 \log n$, $4\sqrt{n}$, $n^{\log_2 n}$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   b) Si dimostri formalmente (cioe' fornendo le costanti) almeno due (a scelta) dei confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, $f_5(n)$, $f_6(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per almeno due diversi indici i.



2. Si supponga di avere una singola risorsa ed alcune richieste di utilizzo, ognuna con una “deadline”, cioe’ un orario entro cui debba essere eseguita. Uno “scheduling” e’ un ordinamento delle richieste; in uno scheduling, si ha un’ “inversione consecutiva” quando una richiesta con deadline d e’ pianificata giusto prima di una richiesta con deadline d', e $d > d'$. Piu’ precisamente, si supponga di avere uno scheduling di n richieste, e sia $D[1..n]$ il vettore in cui $D[i]$, per $i = 1, 2, \cdots, n$, e’ la deadline della richiesta che compare all’i-esimo posto nello scheduling. Un’inversione consecutiva e’ quindi un indice i, con $i = 1, 2, \cdots, n - 1$, per cui $D[i] > D[i + 1]$. Per esempio sia $D[1..n] = [3, 4, 8, 6, 9]$; allora $i = 3$ e’ un’inversione consecutiva.

   (a) Argomentare che, se $D[1] > D[n]$, allora lo scheduling ha almeno un’inversione consecutiva.

   (b) Descrivere un algoritmo che, avendo in input un vettore di interi $D[1..n]$, con $n \geq 2$ e $D[1] > D[n]$, trovi un’inversione consecutiva in tempo $O(\log n)$. Indicare la tecnica di progettazione di algoritmi usata e giustificarne la correttezza. Analizzare il tempo di esecuzione (verificando che sia quello richiesto).

3. a) Descrivere un algoritmo di programmazione dinamica per la seguente variante al problema dello zaino: Dati n oggetti di peso $w_1, w_2, \cdots, w_n$ e valore $v_1, v_2, \cdots, v_n$ ed uno zaino di capacita’ W, trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale e’ al massimo W, con la condizione che non possono essere presi due oggetti con indici consecutivi (ovvero gli oggetti i-esimo ed $(i + 1)$-esimo, per $i = 1, 2, \cdots, n - 1$).

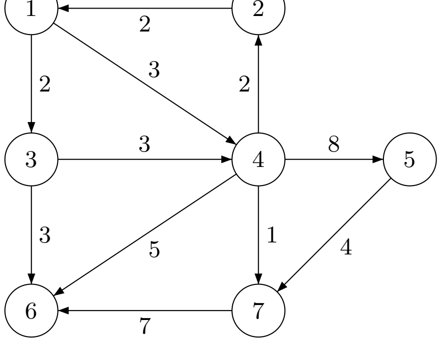
   b) Analizzare la complessita’ di tempo e di spazio dell’algoritmo proposto.

   c) Scrivere lo pseudocodice dell’algoritmo proposto.

4. a) Si definisca il **problema** del Minimo albero di ricoprimento (MST) per un grafo.

   b) Fornire l’esempio di un grafo con due distinti MST.

5. Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l’algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.



17 Appello 30 Gennaio 2012

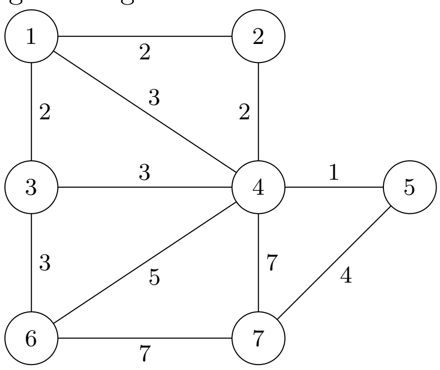
1. Risolvere la seguente relazione di ricorrenza

$$T(n) = 2T(n/2) + n^2$$

supponendo che n sia una potenza di 2 e $T(1) = 3$.

2. Descrivere ed analizzare un algoritmo che dato un array $A[1..n]$ di interi distinti, con $A[1] = x$, permuti l'array in modo che, al termine dell'esecuzione, restituisca un indice q con $1 \leq q \leq n$ in maniera che $A[q] = x$, e per ogni $1 \leq i < q$, si abbia $A[i] < x$, e per ogni $q < i \leq n$, si abbia $A[i] > x$. L'algoritmo deve avere complessita' di tempo $O(n)$. Giustificare la correttezza dell'algoritmo.

3. Eseguire l'algoritmo di Kruskal sul seguente grafo, mostrando gli aggiornamenti effettuati.



4. Si consideri il problema di un cassiere che deve dare il resto di N centesimi avendo a disposizione quante si voglia monete da 1, 5 e 10 centesimi, volendo utilizzare il minimo numero di monete.

   a) Descrivere ed analizzare la complessita' di tempo di un algoritmo 'greedy' che risolva il problema esposto.

   b) Giustificarne la correttezza.

5. a) Dare la definizione di grafo.

   b) Dare la definizione di grafo bipartito.

   c) Dare la definizione di 'matching' in un grafo bipartito.

   d) Dare la definizione di 'matching massimale' in un grafo bipartito.

   e) Definire il problema del 'matching' in un grafo bipartito.

   f) Descrivere l'algoritmo studiato per risolvere il problema del punto e).

18 Appello 16 Febbraio 2012

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 3n^2 + 16 \log n$ e $T_{A'}(n) = T_{A'}(n-1) + 2n$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. a) Definire formalmente il problema della selezione delle attività (*interval scheduling*).

   b) Eseguire l'algoritmo studiato per il problema di cui al punto a) sul seguente insieme di intervalli, giustificando tutti gli aggiornamenti eseguiti:

   $S = \{(1,3),(3,7),(10,12),(1,2),(4,6),(7,10),(1,5),(6,9)\}$.

   c) Indicare tutte le altre (eventuali) soluzioni ottimali.

3. Descrivere ed analizzare un algoritmo che in tempo $O(n + r)$ ordini un array $A[1..n]$ i cui elementi sono interi compresi fra 1 ed r. Commentare l'algoritmo proposto.

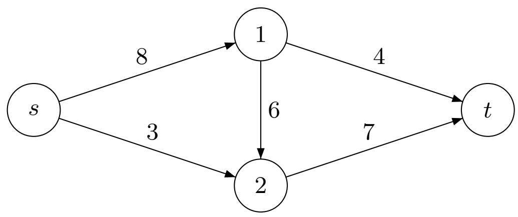
4. a) Descrivere ed analizzare un algoritmo che, dato un grafo (non-orientato) $G = (V, E)$, decida se G e' aciclico oppure no e, in caso non sia aciclico, fornisca un ciclo di G.

   b) Descrivere ed analizzare un algoritmo per lo stesso problema di cui al punto a), nel caso in cui il grafo sia orientato.

5. a) Dare la definizione di "rete di flusso."

   b) Definire il "problema del massimo flusso".

   c) Eseguire l'algoritmo di Ford-Fulkerson sulla seguente rete di flusso, scegliendo ad ogni iterazione, il cammino aumentante di *bottleneck* massimo. Indicare gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto.



19 Appello 18 Aprile 2012

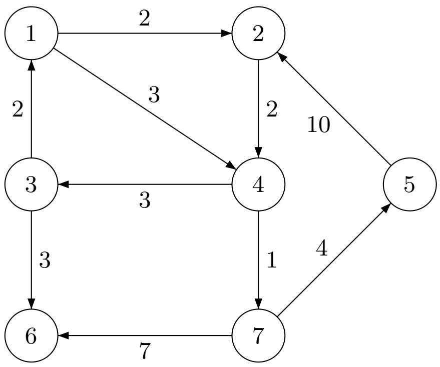
1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 7n^2 + 2\sqrt{n}$ e $T_{A'}(n) = 4T_{A'}(n/2) + n$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. (a) Modificare l'algortimo Partition in modo che, dato un vettore $A[1 \cdots n]$ di interi in input, una volta scelto $A[1]$ come pivot, restituisca un indice q, $1 < q \leq n$, tale che ogni elemento in $A[1 \cdots q-1]$ sia **maggiore** del pivot, mentre ogni elemento in $A[q \cdots n]$ sia **minore o uguale** del pivot. Analizzare la complessità di tempo dell'algoritmo proposto.

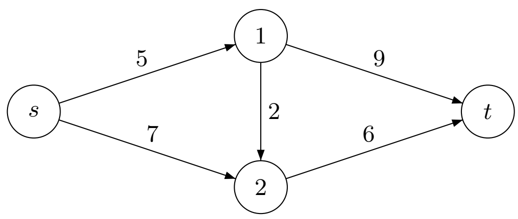
   (b) Descrivere un algoritmo ricorsivo che, dato un vettore $A[1..n]$ di interi in input, lo ordini in ordine **decrescente**, utilizzando l'algoritmo descritto al punto a). Analizzare la complessità di tempo dell'algoritmo proposto.

3. Descrivere ed analizzare un algoritmo per la seguente variante al problema dello zaino: Dati n oggetti di peso $w_1, w_2, \cdots, w_n$ e valore $v_1, v_2, \cdots, v_n$ ed uno zaino di capacita' W, trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale e' al massimo W, con la condizione che ogni oggetto possa essere preso anche piu' di una volta.

4. Sia $G = (V, E)$ il seguente grafo pesato. Eseguire l'algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.



5. Si esegua l'algoritmo di Ford- Fulkerson sulla seguente rete di flusso, indicando gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto ed un taglio di capacita' minima.



20 Appello 28 Giugno 2012

1. Si considerino le seguenti funzioni: $\sqrt{3^n}$, $\log(2n+1)$, 2^n, $2\log n + 1$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   b) Si dimostri formalmente (cioe' fornendo le costanti) tutti i confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per ogni indice i, $1 \leq i \leq 3$.

2. Descrivere ed analizzare un algoritmo che presi in input un vettore $A[1..n]$ di interi distinti e un intero i, $2 \leq i \leq n$, tale che $A[1..i-1]$ e' ordinato in ordine crescente, inserisca $A[i]$ nell'ordine che gli compete fra i suoi predecessori, ovvero modifichi A in maniera che dopo l'esecuzione $A[1..i]$ contenga gli stessi elementi di prima, ma in ordine crescente, e i restanti elementi di A siano immutati. L'algoritmo deve avere una complessita' di tempo $O(logn)$.

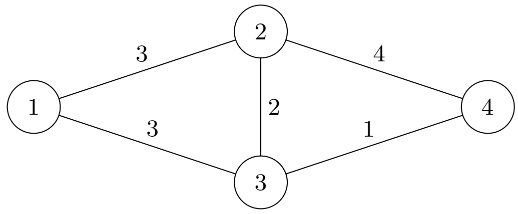
   Per esempio: se $A[1..6] = [2, 4, 7, 3, 8, 1]$ ed $i = 4$, al termine dell'esecuzione il vettore risultante sara' $A[1..6] = [2, 3, 4, 7, 8, 1]$.

3. a) Si definisca il problema della selezione delle attivita' (*interval scheduling*).

   b) Si consideri l'algoritmo *greedy* che risolve il problema scegliendo di volta in volta l'attivita' che inizia per ultima. Dimostrare la correttezza dell'algoritmo, mostrando che la soluzione fornita e' sempre ottimale. Non e' necessario scrivere lo pseudocodice dell'algoritmo.

4. Descrivere ed analizzare un algoritmo che, dati un grafo $G = (V, E)$, e un arco $(u, v) \in E$, restituisca Vero o Falso secondo che esista o meno un minimo albero di ricoprimento (*minimum spanning tree*) di G che non contiene (u, v).

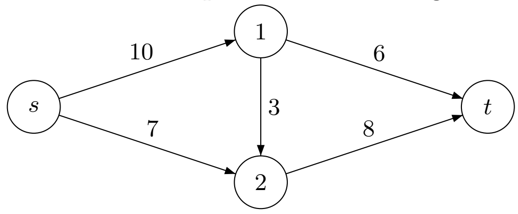
   Mostrare infine l'esecuzione dell'algoritmo proposto sul grafo seguente e arco $(1, 2)$.



5. (a) Si definisca cos'e' una rete di flusso.

   (b) Si definisca il problema del flusso massimo.

   (c) Si esegua l'algoritmo di Ford-Fulkerson sulla seguente rete di flusso, indicando gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto, un'assegnazione di flusso sulla rete di partenza ed un taglio di capacita' minima.



21 Appello 13 Luglio 2012

1. Risolvere la seguente relazione di ricorrenza

$$T(n) = 3T(n/2) + n$$

   supponendo che n sia una potenza di 2 e $T(2) = 2$.

2. Si consideri il problema della moltiplicazione di due interi espressi in notazione binaria (in base 2), dove la complessita' di tempo e' espressa in termini di operazioni aritmetiche su bit.

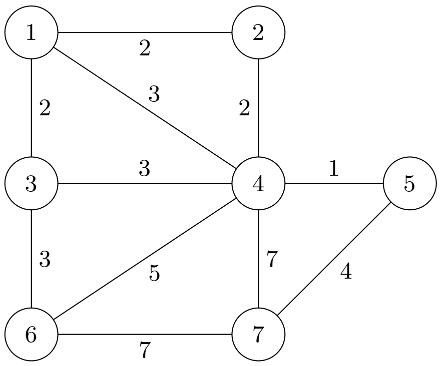
   a) Qual e' la complessita' di tempo dell'algoritmo per la moltiplicazione di interi come studiato alle scuole elementari?

   b) Descrivere ed analizzare un algoritmo per la moltiplicazione di due interi rappresentati in binario, che sia basato sulla tecnica divide-et-impera e si fondi sull'osservazione che un intero x rappresentato con n bit si puo' decomporre in $x = x_1 \cdot 2^{n/2} + x_2$, per opportuni valori degli interi x_1 e x_2. Si supponga che n sia una potenza di 2. L'algoritmo deve essere il piu' efficiente possibile.

   Esempio $x = 1101 = x_1\dot{2}^2 + x_2$ con $x_1 = 11$ e $x_2 = 01$; $y = 1010 = y_1\dot{2}^2 + y_2$ con $y_1 = y_2 = 10$.

3. a) Eseguire l'algoritmo di Kruskal sul seguente grafo G, mostrando gli aggiornamenti effettuati.

   b) Completare poi il grafo G con archi di peso 10 e risolvere il problema del cluster con $k = 3$.



4. Lungo un fiume ci sono n approdi. A ciascuno di questi approdi é possibile fittare una canoa che puó essere restituita ad un altro approdo. E' praticamente impossibile andare controcorrente. Il costo del fitto di una canoa da un punto di partenza i ad un punto di arrivo j, con $i < j$, é denotato con $C(i, j)$. E' possibile che per andare da i a j sia piú economico effettuare alcune soste e cambiare la canoa piuttosto che fittare una unica canoa. Se si fitta una nuova canoa in $k_1 < k_2 < ... < k_l$ allora il costo totale per il fitto é $C(i, k_1)+C(k_1, k_2)+...+C(k_{l-1}, k_l)+C(k_l, j)$.

   a) Descrivere un algoritmo che dati in input i costi $C(i, j)$, determini il **costo** minimo per recarsi da 1 ad n. Analizzare la complessita' dell'algoritmo proposto.

   b) Descrivere un algoritmo che dati in input i costi $C(i, j)$, determini anche un **cammino** di costo minimo per recarsi da 1 ad n. Analizzare la complessita' dell'algoritmo proposto.

   Esempio Sia $n = 4$ e costi $C(1, 2) = 1$, $C(1, 3) = 2$, $C(1, 4) = 4$, $C(2, 3) = 1$, $C(2, 4) = 1$, $C(3, 4) = 1$.

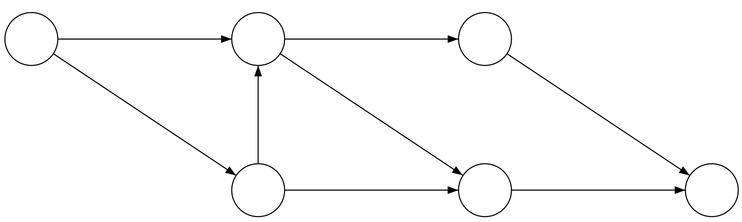
   Allora i possibili modi per andare da 1 a 4 sono: $(1, 4)$, $(1, 2, 4)$, $(1, 3, 4)$, $(1, 2, 3, 4)$. I rispettivi costi sono: 4, 2, 3, 3. Il costo minimo e' quindi 2. Un cammino di costo minimo e': $(1, 2, 4)$.

5. a) Definire cos'e' un ordinamento topologico per un grafo orientato.

   b) Quando un grafo orientato ammette un ordinamento topologico?

   c) Fornire l'esempio di un grafo orientato che non ammette ordinamento topologico.

   d) Fornire un ordinamento topologico per il seguente DAG (*Directed Acyclic Graph*), utilizzando l'algoritmo studiato e spiegando le operazioni via via eseguite.



22 Appello 13 Settembre 2012

1. a) Si risolva la seguente relazione di ricorrenza, supponendo che n sia una potenza di 3, che $T(1) = 1$ e senza far ricorso al Master Theorem:

$$T(n) = 3T(n/3) + 1$$

   b) Si risolva la seguente relazione di ricorrenza, supponendo che $b > 1$, che n sia una potenza di b, che $T(1) = 1$ e senza far ricorso al Master Theorem:

$$T(n) = bT(n/b) + 1$$

2. Descrivere un algoritmo che dati un vettore $A[1..n]$ di interi e un intero k, con $1 \leq k \leq n$, restituisca un vettore $B[1..k]$ con i k interi piu' grandi in ordine crescente. L'algoritmo deve essere iterativo e mantenere nelle successive iterazioni in $B[1..k]$ i k interi piu' grandi di $A[1..i]$ in ordine crescente ($i = k, ..., n$).

   Analizzare la complessita' dell'algoritmo proposto.

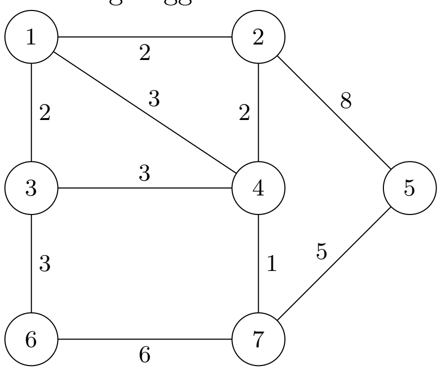
3. a) Definire il problema dello zaino (versione 0-1).

   b) Descrivere ed analizzare un algoritmo di programmazione dinamica che lo risolve (bisogna mostrare la relazione di ricorrenza che ne descrive una soluzione e lo pseudocodice che la implementa).

4. a) L'algoritmo di Prim per il calcolo di un MST di un grafo si basa sulla proprieta' del taglio; enunciare tale proprieta'.

   b) Sia $G = (V, E)$ il seguente grafo. Spiegare perche' l'arco $(1, 2)$ deve necessariamente appartenere ad ogni MST di G. Spiegare perche' anche gli archi $(1, 3)$ e $(2, 4)$ devono necessariamente appartenere ad ogni MST di G. *Suggerimento*: usare la proprieta' del taglio.

   c) Eseguire l'algoritmo di Prim sul seguente grafo $G = (V, E)$ partendo dal nodo $s = 1$ e mostrando gli aggiornamenti effettuati.



5. a) Esibire un grafo $G = (V, E)$ con 6 vertici e un vertice $v \in V$, tale che l'albero ottenuto con una visita BFS di G a partire da v abbia altezza 1, mentre l'albero ottenuto con una visita DFS di G a partire da v abbia altezza 5.

   b) Esibire una famiglia di grafi $G_n = (V_n, E_n)$ con n vertici, $n = 2, 3, ...,$ e un vertice $v \in V_n$, tale che l'albero ottenuto con una visita BFS di G_n a partire da v abbia altezza 1, mentre l'albero ottenuto con una visita DFS di G a partire da $v \in V_n$ abbia altezza $n - 1$. Si puo' generalizzare l'esempio fornito al punto a).

23 Appello 5 Novembre 2012

1. Senza far ricorso al Master Theorem, si risolva la seguente relazione di ricorrenza, supponendo che n sia una potenza di 2 e che $T(1) = 1$:

$$T(n) = 4T(n/2) + 3n$$

2. Descrivere ed analizzare un algoritmo basato sul paradigma *divide et impera* che dato un vettore *ordinato* $A[1..n]$ di interi strettamente positivi (cioe' per ogni $1 \leq i \leq n$, $A[i] \geq 1$), restituisca il numero di occorrenze di 1 nel vettore A. L'algoritmo deve avere complessita' di tempo $O(\log n)$.

3. a) Definire il problema della ricerca dei cammini minimi in un grafo.

   b) Descrivere l'algoritmo di Bellman e Ford per la soluzione del problema di cui al punto a).

4. Si consideri un grafo pesato con costi (a due a due) distinti sugli archi.

   a) Enunciare la proprieta' del taglio.

   b) Dimostrare che l'arco di peso minimo del grafo, appartiene a qualsiasi minimum spanning tree del grafo.

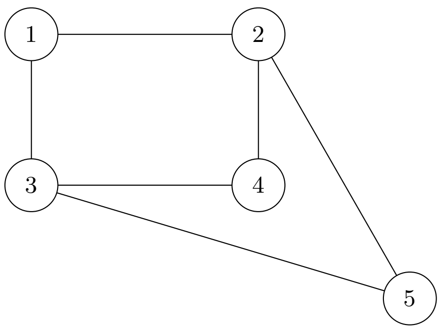
5. a) Definire cos'e' un grafo bipartito.

   b) Definire cos'e' un matching massimale in un grafo bipartito.

   Si consideri il seguente grafo G.

   c) Dimostrare che G e' bipartito, eseguendo l'algoritmo di test studiato.

   d) Determinare un matching massimale di G, eseguendo l'algoritmo studiato.



24 Appello 8 Gennaio 2013

1. Risolvere la seguente relazione di ricorrenza

$$T(n) = 3T(n/3) + n^2$$

   supponendo che n sia una potenza di 3 e $T(1) = 3$.

2. Progettare ed analizzare un algoritmo basato sulla tecnica Divide et Impera che, dato un vettore $A[1...n]$, con $A[i] \in \{0, 1\}$, per ogni $i = 1, ..., n$, restituisca VERO se esiste $i \in \{1, ..., n-1\}$ tale che $A[i] = 1$ e $A[i+1] = 0$, FALSO altrimenti.

3. Lungo un fiume ci sono n approdi. A ciascuno di questi approdi é possibile fittare una canoa che puó essere restituita ad un altro approdo. E' praticamente impossibile andare controcorrente. Il costo del fitto di una canoa da un punto di partenza i ad un punto di arrivo j, con $i < j$, é denotato con $C(i, j)$. E' possibile che per andare da i a j sia piú economico effettuare alcune soste e cambiare la canoa piuttosto che fittare una unica canoa. Se si fitta una nuova canoa in $k_1 < k_2 < ... < k_l$ allora il costo totale per il fitto é $C(i, k_1)+C(k_1, k_2)+...+C(k_{l-1}, k_l)+C(k_l, j)$.

   Descrivere un algoritmo che dato in input i costi $C(i, j)$, e un intero f, $0 \leq f \leq n-2$, determini il costo minimo per recarsi da 1 ad n, effettuando esattamente f **fermate intermedie**. Analizzare la complessità dell'algoritmo proposto.

   Esempio. Sia $n = 4$, e $C(1, 2) = 1$, $C(1, 3) = 2$, $C(1, 4) = 4$, $C(2, 3) = 1$, $C(2, 4) = 1$, $C(3, 4) = 1$. Allora i possibili modi per andare da 1 a 4 sono: $1 \to 4$ con 0 fermate intermedie, $1 \to 2 \to 4$ e $1 \to 3 \to 4$ con 1 fermata intermedia, $1 \to 2 \to 3 \to 4$ con 2 fermate intermedie. I rispettivi costi sono: 4, 2, 3, 3.

4. Sia $G = (V, E)$ un grafo diretto. Dimostrare almeno due delle seguenti affermazioni.

   a) Se G e' aciclico allora G contiene un vertice senza archi **uscenti**.

   b) Mostrare con un controesempio che il viceversa del punto a) e' falso.

   c) Se G e' aciclico e contiene un vertice senza archi **uscenti** allora G ha un ordinamento topologico.

   d) Se G ha un ordinamento topologico allora G e' aciclico.

5. a) Si supponga di rappresentare un grafo diretto e pesato $G = (V, E)$, con $V = \{1, 2, \cdots, n\}$, tramite una matrice M di dimensione $n \times n$, a coefficienti interi, tale che: esiste un arco $(i, j) \in E$ se e solo se $M[i, j] \neq 0$, e in tal caso il peso dell'arco (i, j) e' proprio il valore $M[i, j]$.

   Si disegni il grafo $G = (V, E)$ rappresentato dalla seguente matrice:

0	1	0	0	0	5	0
1	0	2	0	0	0	0
0	0	0	1	3	0	0
0	0	0	0	5	2	3
0	0	0	0	0	0	0
0	0	0	4	0	0	4
0	0	0	0	1	0	0

   b) Si esegua l'algoritmo di Dijkstra su G con $s = 1$, mostrando gli aggiornamenti effettuati e indicando per ogni vertice il cammino minimo che lo collega ad 1 e il suo costo.

25 Appello 24 Gennaio 2013

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = \sqrt{n} \log n$ e $T_{A'}(n) = 2T_{A'}(n/2) + 1$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. Si consideri il problema di determinare simultaneamente il minimo ed il massimo in un vettore di interi. Si descriva un algoritmo basato sulla tecnica Divide et Impera che, dato un vettore $A[1..n]$ di interi, restituisca una coppia (min, max), dove min e' il valore minimo in A e max il suo valore massimo. Analizzare il tempo di esecuzione.

3. Si supponga di dover effettuare un viaggio dalla localita' A alla localita' B con una moto con un'autonomia di 100 chilometri a serbatoio pieno. Lungo il percorso da A a B si incontrano n distributori di carburante. Sia d_0 la distanza da A al primo distributore, d_i la distanza che separa il distributore i dal distributore $i+1$, per $i = 1, 2, \cdots, n-1$ e d_n la distanza dell'ultimo distributore da B; inoltre d_i e' inferiore a 100 chilometri, per ogni $i = 0, 1, \cdots, n$.

   a) Quale puo' essere un criterio di scelta per minimizzare il numero di distributori in cui fermarsi? (*Suggerimento: in quale distributore conviene fermarsi la prima volta?*)

   b) Descrivere ed analizzare un algoritmo greedy basato sul criterio proposto.

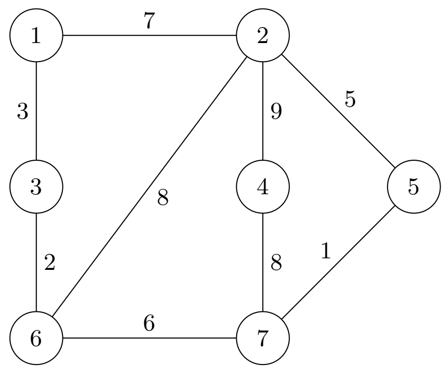
   c) Argomentare la correttezza dell'algoritmo proposto.

4. Sia G il grafo seguente.

   a) Eseguire l'algoritmo di Kruskal su G

   b) Si consideri il grafo G^c ottenuto aggiungendo a G tutti gli archi mancanti affinche' esso sia un grafo completo; si assegni a tali archi costo 10 (non occorre disegnarlo).

   Trovare un k-clustering per G^c con $k = 3$.



5. Si consideri il problema dell'allineamento di due stringhe (*sequence alignment*). Eseguire l'algoritmo di programmazione dinamica studiato per calcolare l'allineamento ottimo fra le stringhe X=NEVE ed Y=VENE, supponendo che la penalita' per un *gap* sia $\delta = 2$, mentre il costo di un *mismatch* sia $\alpha_{pq} = 3$ per ogni coppia di caratteri p e q distinti.

   Puo' essere utile ricordare che la relazione di ricorrenza che permette di risolvere il problema e' la seguente: $OPT(i, j) = min\{\alpha_{x_i y_j} + OPT(i-1, j-1), \delta + OPT(i-1, j), \delta + OPT(i, j-1)\}$, con le opportune condizioni iniziali.

26 Appello 13 Febbraio 2013

1. Si considerino le seguenti funzioni: $(\sqrt{n}+1)^2$, $2^{\log_2 n+3}$, 2^{n+1}, $n^{\log_2 n}$, 2^{2n}, n^{15}.

   Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   Dimostrare formalmente la validita' di almeno due confronti fra una funzione e la seguente nell'ordine proposto.

2. Descrivere **due** diversi algoritmi che verificano se un vettori $A[1..n]$ di interi e' una permutazione di $\{1, ..., n\}$. Giustificarne la correttezza. Confrontare la complessita' di tempo e di spazio dei due algoritmi proposti.

3. Descrivere ed analizzare un algoritmo di programmazione dinamica per risolvere il seguente problema: dati un insieme di n oggetti $S = \{1, 2, ..., n\}$ e un intero C, tali che ad ogni oggetto i, per $i = 1, 2, ..., n$, sia assegnato un costo intero non-negativo c_i, selezionare un sottoinsieme di S, in modo che la somma dei costi degli oggetti selezionati sia la massima possibile senza superare C.

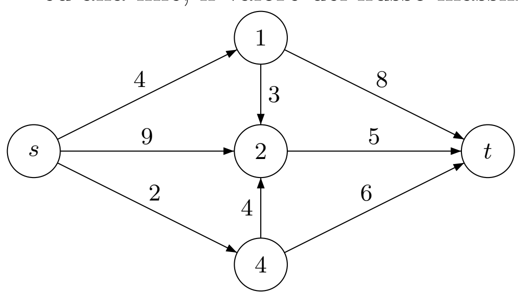
4. Si consideri l'algoritmo studiato per calcolare un ordinamento topologico in un DAG.

   Si estenda tale algoritmo in modo che, dato un grafo diretto G, restituisca: a) un ordinamento topologico, se G e' un DAG, b) un ciclo di G, se G non e' un DAG.

   Giustificare la correttezza dell'algoritmo proposto, menzionando i risultati teorici su cui si basa.

5. a) Dare la definizione di rete di flusso e del problema del flusso.

   b) Eseguire l'algoritmo di Ford-Fulkerson sulla seguente rete di flusso, scegliendo ad ogni iterazione il cammino con *bottleneck* massimo. E' necessario indicare gli aggiornamenti effettuati, ed alla fine, il valore del flusso massimo ottenuto ed un taglio di capacita' minima.



27 Appello 19 Aprile 2013

1. Si risolva la seguente relazione di ricorrenza, supponendo per semplicita' che n sia una potenza di 2 e che $T(1) = 1$:
$$T(n) = 4T(n/2) + 3n$$

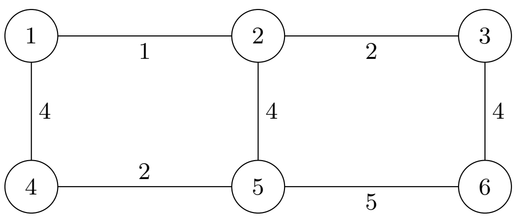
2. Descrivere ed analizzare un algoritmo basato sul paradigma *divide et impera* che dato un vettore *ordinato* $A[1..n]$ di interi strettamente positivi (cioe' per ogni $1 \leq i \leq n$, $A[i] \geq 1$), restituisca il numero di occorrenze di 1 nel vettore A. L'algoritmo deve avere complessita' di tempo $O(\log n)$.

3. a) Definire il problema della ricerca dei cammini minimi in un grafo.

   b) Descrivere l'algoritmo di Bellman e Ford per la soluzione del problema di cui al punto a).

4. a) Descrivere ed analizzare un algoritmo che, dati un grafo pesato $G = (V, E)$, ed un arco $(u, v) \in E$, decida se esiste in G un albero minimo di copertura (MST) che contiene l'arco (u, v).

   b) Eseguire l'algoritmo proposto sul seguente grafo $G = (V, E)$ ed arco $(1, 4)$.



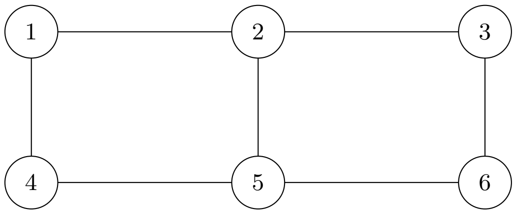
5. a) Definire cos'e' un grafo bipartito.

   b) Definire cos'e' un matching massimale in un grafo bipartito.

   Si consideri il seguente grafo G.

   c) Dimostrare che G e' bipartito, eseguendo l'algoritmo di test studiato.

   d) Determinare un matching massimale di G, eseguendo l'algoritmo studiato.



28 Appello 25 Giugno 2013

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 5n^2 + 11 \log n$ e $T_{A'}(n) = \sqrt{n^3} \log n + n$, quale dei due algoritmi è asintoticamente più efficiente in termini di tempo? È necessario giustificare la risposta.

2. Uno studente universitario ha segnato i suoi n voti riportati finora in un array $V[1..n]$. I voti sono interi compresi fra 18 e 30.

   Descrivere un algoritmo che dato l'array $V[1..n]$, lo permuti in modo che tutti i voti maggiori di 24 si presentino prima di quelli minori o uguali a 24 e restituisca l'indice m, $0 \leq m \leq n$, tale che $V[i] > 24$ per ogni $1 \leq i \leq m$ e $V[i] \leq 24$ per ogni $m < i \leq n$. L'algoritmo deve avere complessità di tempo $O(n)$ e deve funzionare *in loco*, cioè non utilizzare altro spazio di memoria oltre all'array V. Giustificarne la correttezza.

3. Cappuccetto Rosso, come sappiamo tutti, deve andare dalla nonna malata a portarle il pranzo e sceglie di attraversare il bosco per poterle cogliere un bel mazzetto di fiorellini. Nel bosco ci sono tanti sentieri possibili che si incrociano in varie aree di sosta con delle belle panchine per potersi riposare un po'. In effetti ci sono n aree di sosta: A_1, A2, ..., A_n, e Cappuccetto Rosso per andare da A_1 ad A_n puo' scegliere fra tutti i sentieri che collegano A_i con A_j con $1 \leq i < j \leq n$. Cappuccetto sa inoltre che sul sentiero da A_i ad A_j potrà raccogliere $f_{i,j}$ fiorellini. Volete aiutare Cappuccetto Rosso ad andare da A_1 ad A_n raccogliendo il numero **massimo** di fiorellini? Si supponga inoltre che il Lupo stia dormendo della grossa e non si accorga nemmeno della presenza della fanciulla nel bosco.

4. Si consideri un grafo pesato G.

   a) Definire cos'e' un minimo albero di ricoprimento (MST) in G.

   b) Enunciare la proprietà del taglio.

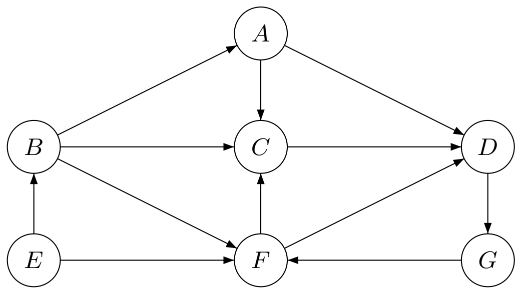
   c) Dimostrare la proprietà del taglio per un grafo pesato con costi a due a due distinti sugli archi.

5. a) Definire cos'è un DAG.

   b) Definire cos'è un ordinamento topologico in un grafo orientato.

c) Descrivere un algoritmo che dato un grafo orientato G, decida se G è un DAG oppure no. Giustificarne la correttezza.

d) Eseguire l'algoritmo descritto al punto c) sul seguente grafo.



29 Appello 11 Luglio 2013

1. Risolvere la seguente relazione di ricorrenza

$$T(n) = 2T(n/2) + n^2$$

supponendo che n sia una potenza di 2 e $T(1) = 3$.

2. a) Sia $C = \{c_1, c_2, \cdots, c_n\}$ un insieme di caratteri ad ognuno dei quali e' associata una frequenza $f(c_i)$, per ogni $i = 1, 2, \cdots, n$. Sia $\gamma : C \to \{0, 1\}$ una codifica binaria di C. Definire cos'e' la lunghezza media per bit di γ, $ABL(\gamma)$ dall'inglese *Average Bit Length*.

   b) Sia adesso $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 15$, $f[b] = 30$, $f[c] = 7$, $f[d] = 20$, $f[e] = 24$, $f[g] = 4$.

   b1) Descrivere una codifica binaria γ_1 di C a lunghezza fissa 3 e calcolare $ABL(\gamma_1)$.

   b2) Descrivere una codifica binaria γ_2 di C che minimizzi la lunghezza media per bit (utilizzando l'algoritmo studiato) e calcolare $ABL(\gamma_2)$.

3. a) Definire il problema della selezione delle attivita' pesate (*Weighted Interval Scheduling*).

   b) Siano dati n intervalli con i rispettivi pesi. Si definisca, per ogni $j = 1, 2, \cdots, n$, $OPT(j)$ come il peso massimo di un insieme di intervalli compatibili, scelti fra i primi j. Si scriva la relazione di ricorrenza soddisfatta da $OPT(j)$, su cui si basa l'algoritmo di programmazione dinamica studiato. Si giustifichi la risposta.

   c) Si considerino adesso i seguenti intervalli: $i_1 = [1, 4]$, $i_2 = [2, 6]$, $i_3 = [5, 7]$, $i_4 = [3, 8]$, $i_5 = [7, 9]$, $i_6 = [8, 10]$, con pesi $v_1 = 2$, $v_2 = 4$, $v_3 = 4$, $v_4 = 7$, $v_5 = 2$, $v_6 = 1$.

   L'array che rappresenta per tale istanza $OPT(j)$, con $j = 0, 1, \cdots, 6$, e' il seguente: $[0, 2, 4, 6, 7, 8, 8]$.

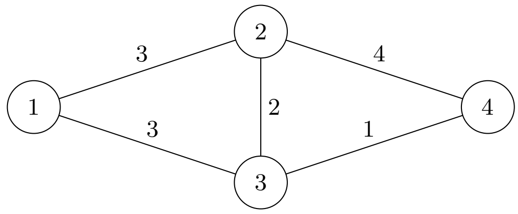
   Analizzando tale array, ricostruire un insieme di intervalli compatibili di peso massimo per gli intervalli dati. Giustificare la risposta.

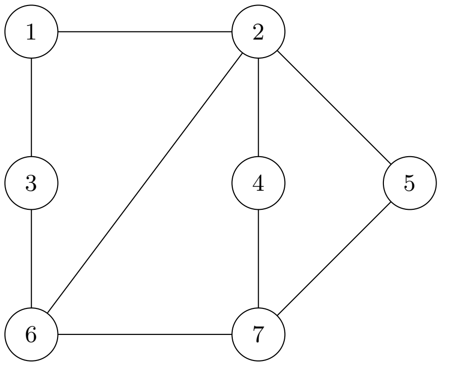
4. Descrivere ed analizzare un algoritmo che, dati un grafo $G = (V, E)$, e un arco $(u, v) \in E$, restituisca VERO o FALSO secondo che esista o meno un minimo albero di ricoprimento (*minimum spanning tree*) di G che non contiene (u, v).

   Mostrare l'esecuzione dell'algoritmo proposto sul grafo seguente e arco $(1, 2)$.

5. Sia G il grafo seguente:

   a) Eseguire la BFS su G a partire dal nodo 1, mostrando i "layers" (strati) ottenuti e l'albero BFS risultante.





b) Dire se G e' un grafo bipartito, motivando la risposta e, in caso affermativo, fornire la partizione dei vertici per cui G e' un grafo bipartito.

c) Trovare un matching massimale in G eseguendo l'algoritmo studiato.

30 Appello 17 Settembre 2013

1. Si considerino le seguenti funzioni: $\sqrt{3^n}$, $\log(2n+1)$, 2^n, $2\log n + 1$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$.

   b) Si dimostrino formalmente (cioe' fornendo le costanti) tutti i confronti affermati al punto a). In altre parole se l'ordine proposto e': $f_1(n)$, $f_2(n)$, $f_3(n)$, $f_4(n)$, allora occorre dimostrare che $f_i(n) = O(f_{i+1}(n))$ per ogni indice i, $1 \leq i \leq 3$.

2. Sia dato un vettore di n elementi che possono assumere solo i tre valori VERDE, BIANCO e ROSSO. Descrivere ed analizzare un algoritmo basato sulla tecnica Divide-et Impera che decida se nel vettore vi sono 3 occorrenze consecutive di ROSSO.

3. a) Definire il problema della selezione delle attivita' pesate (*Weighted Interval Scheduling*).

   b) Siano dati n intervalli con i rispettivi pesi. Si definisca, per ogni $j = 1, 2, \cdots, n$, $OPT(j)$ come il peso massimo di un insieme di intervalli compatibili, scelti fra i primi j. Si scriva la relazione di ricorrenza soddisfatta da $OPT(j)$, su cui si basa l'algoritmo di programmazione dinamica studiato. Si giustifichi la risposta.

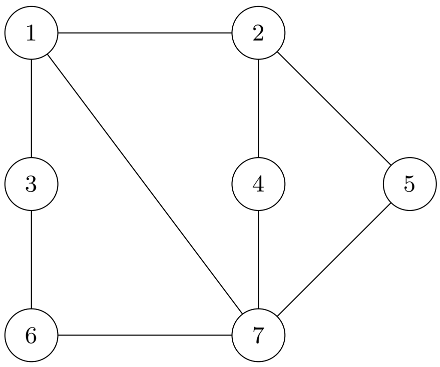
   c) Si considerino adesso i seguenti intervalli: $i_1 = [1,4]$, $i_2 = [2,6]$, $i_3 = [4,7]$, $i_4 = [3,8]$, $i_5 = [7,9]$, $i_6 = [8,9]$, con pesi $v_1 = 2$, $v_2 = 2$, $v_3 = 4$, $v_4 = 7$, $v_5 = 2$, $v_6 = 3$.

   L'array che rappresenta per tale istanza $OPT(j)$, con $j = 0, 1, \cdots, 6$, e' il seguente: $[0, 2, 4, 6, 7, 8, 10]$.

   Analizzando tale array, ricostruire un insieme di intervalli compatibili di peso massimo per gli intervalli dati. Giustificare la risposta.

4. La *larghezza* di un albero (radicato) e' il numero massimo di nodi che si trovano allo stesso livello (o profondita'). Si descriva tramite pseudocodice un algoritmo efficiente per calcolare la larghezza di un albero.

5. Sia G il grafo seguente:



a) Eseguire la BFS su G a partire dal nodo 1, mostrando i "layers" (strati) ottenuti e l'albero BFS risultante.

b) Dire se G e' un grafo bipartito, motivando la risposta e, in caso affermativo, fornire la partizione dei vertici per cui G e' un grafo bipartito.

c) Trovare un matching massimale in G eseguendo l'algoritmo studiato.

31 Appello 4 Novembre 2013

1. Descrivere ed analizzare un algoritmo per la seguente variante al problema dello ziano: Dati n oggetti di peso $w_1, w_2, \cdots, w_n$ e valore $v_1, v_2, \cdots, v_n$ ed uno zaino di capacita' W, trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale e' al massimo W, con la condizione che ogni oggetto puo' essere preso al massimo 2 volte.

2. Progettare ed analizzare un algoritmo basato sulla tecnica Divide et Impera che, dato un vettore *ordinato* $A[1...n]$, ed un intero k, restituisca VERO se esiste $i \in \{1, ..., n\}$ tale che $A[i] > k$, FALSO altrimenti, in tempo $O(\log n)$.

3. a) Descrivere tramite pseudocodice un algoritmo *greedy* (goloso) per il problema della selezione delle attivita' che sia basato sulla scelta dell'attivita' che **inizia per ultima**.

   b) Analizzarne il tempo di esecuzione.

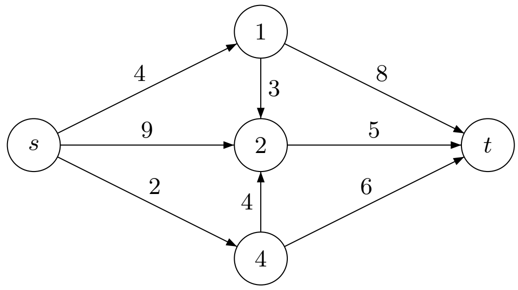
   c) Giustificarne la correttezza.

4. a) Descrivere tramite pseudocodice ed analizzare la complessita' di tempo di un algoritmo di programmazione dinamica per il calcolo dei numeri di Fibonacci in tempo $O(n)$. L'esercizio potra' avere il punteggio massimo solo se lo spazio di memoria e' costante ($O(1)$).

   b) Evidenziare gli elementi che fanno dell'algoritmo proposto al punto a), un algoritmo di programmazione dinamica.

5. a) Dare la definizione di rete di flusso e del problema del flusso.

   b) Eseguire l'algoritmo di Ford-Fulkerson sulla seguente rete di flusso, scegliendo ad ogni iterazione il cammino con *bottleneck* **massimo**. E' necessario indicare gli aggiornamenti effettuati, ed alla fine, il flusso massimo ottenuto con il suo valore, ed un taglio di capacita' minima.



32 Appello 13 Gennaio 2014

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = \sqrt{n}\log n$ e $T_{A'}(n) = T_{A'}(n-1) + 3$ con $T_{A'}(1) = 1$, quale dei due algoritmi e' asintoticamente piu' efficiente in termini di tempo? E' necessario giustificare la risposta.

2. a) Definire formalmente il problema dello *scheduling che minimizza il ritardo.*

   b) Considerare la variante del problema in cui la soluzione cercata, anziche' lo scheduling ottimale, e' il suo ritardo. Descrivere tramite pseudocodice ed analizzare un algoritmo che risolve tale variante al problema.

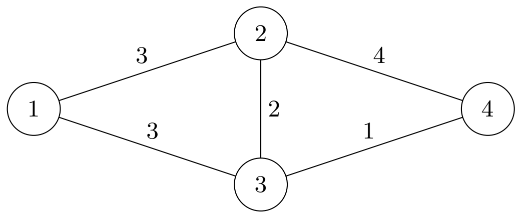
3. a) Definire formalmente il problema dello zaino.

   b) Eseguire l'algoritmo di programmazione dinamica studiato per il problema di cui al punto a) sul seguente insieme di dati, calcolando oltre al valore ottimo, anche una soluzione otttimale al problema. E' necessario mostrare e commentare i passi salienti dell'esecuzione.

   $\{1, 2, 3\}$, $w_1 = w_2 = 2$, $w_3 = 3$; $v_1 = 4$, $v_2 = 2$, $v_3 = 5$ e $W = 5$.

4. Descrivere ed analizzare un algoritmo che, dati un grafo $G = (V, E)$, e un arco $(u, v) \in E$, restituisca VERO o FALSO secondo che esista o meno un minimo albero di ricoprimento (*minimum spanning tree*) di G che **non** contiene (u, v). Commentare l'algoritmo.

   Mostrare l'esecuzione dell'algoritmo proposto sul grafo seguente e arco $(1, 2)$.

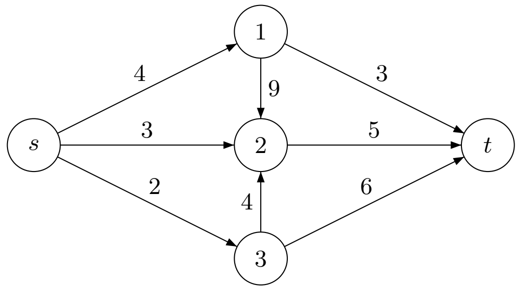


5. a) Dare la definizione di rete di flusso e del problema del flusso.

   b) Eseguire l'algoritmo di Ford-Fulkerson sulla seguente rete di flusso, scegliendo ad ogni iterazione il cammino con *bottleneck* massimo. E' necessario indicare gli aggiornamenti effettuati, ed alla fine il valore del flusso massimo ottenuto ed un taglio di capacita' minima.

33 Appello 5 Febbraio 2014

1. Si consideri il seguente algoritmo, si esprima la complessità di tempo dell'algoritmo mediante una relazione di ricorrenza e se ne dia una soluzione.




```
ALGORITMO(n)
if n = 1 then return 0
         else ALGORITMO(n-1)
              x <-- 0
              for i=1,..,n do
                  x <--x + 2i
```

2. a) Sia $C = \{c_1, c_2, \cdots, c_n\}$ un insieme di caratteri ad ognuno dei quali e' associata una frequenza $f(c_i)$, per ogni $i = 1, 2, \cdots, n$. Sia $\gamma : C \rightarrow \{0, 1\}$ una codifica binaria di C. Definire cos'e' la lunghezza media per bit di γ, $ABL(\gamma)$ dall'inglese *Average Bit Length*.

   b) Sia adesso $C = \{a, b, c, d, e, g\}$ con le seguenti frequenze $f[a] = 25$, $f[b] = 28$, $f[c] = 7$, $f[d] = 10$, $f[e] = 26$, $f[g] = 4$.

   b1) Descrivere una codifica binaria γ_1 di C a lunghezza fissa 3 e calcolare $ABL(\gamma_1)$.

   b2) Descrivere una codifica binaria γ_2 di C che minimizzi la lunghezza media per bit (utilizzando l'algoritmo studiato) e calcolare $ABL(\gamma_2)$.

3. Dato un grafo diretto aciclico $G = (V, E)$, descrivere un algoritmo di **programmazione dinamica** che associ ad ogni nodo $u \in V$ il suo *grado di raggiungibilità*, ovvero il numero di nodi raggiungibili da u (si noti che u è raggiungibile da se stesso). E' necessario commentare il funzionamento dell'algoritmo. L'esercizio potrà avere il massimo della valutazione solo se contiene anche lo pseudocodice e l'analisi dei costi computazionali.

4. a) Definire la rappresentazione di un grafo orientato con matrice di adiacenza.

   b) Descrivere ed analizzare un algoritmo che, data la rappresentazione con matrice di adiacenza di un grafo orientato $G = (V, E)$, con $|V| = n$, determini se G contiene un *pozzo universale*, ovvero un nodo $p \in V$ senza archi uscenti e con $n - 1$ archi entranti.

   Nota Un bonus supplementare verrà dato se l'algoritmo ha un tempo di esecuzione $\Theta(n)$.

5. a) Definire cos'e' un grafo bipartito.

   b) Definire cos'e' un matching massimale in un grafo bipartito.

   Si consideri il seguente grafo G.

   c) Dimostrare che G e' bipartito, eseguendo l'algoritmo di test studiato.

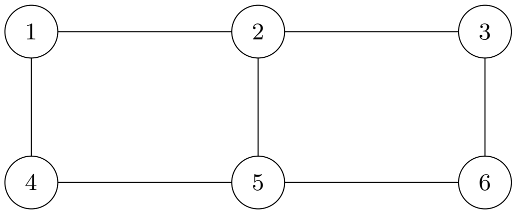
   d) Determinare un matching massimale di G, eseguendo l'algoritmo studiato.

34 Appello 24 Febbraio 2014

1. Si considerino le seguenti funzioni: $\log^3 n + \sqrt{n}$, $\log n$, $2^{\sqrt{n}}$, $n^{\log n}$, n^3, $2^{3 \log n}$.

   a) Si ordinino le funzioni scrivendole da sinistra a destra, in modo tale che la funzione $f(n)$ sia posta a sinistra di $g(n)$ se $f(n) = O(g(n))$, precisando anche se $f(n) = \Theta(n)$, oppure no.

   b) Giustificare le risposte.



2. Uno studente universitario ha segnato i suoi n voti finora riportati agli esami in un array $V[1..n]$. I voti sono interi compresi fra 18 e 30 e adesso vuole suddividerli in tre gruppi: i voti (per lui) "bassi" fra 18 e 22, i voti "medi" da 23 e 26, e i voti "alti" da 28 a 30.

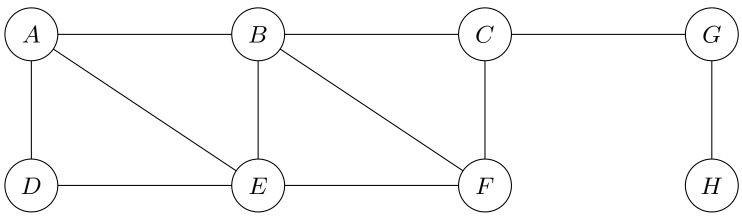
   Descrivere un algoritmo che, dato l'array $V[1..n]$, lo permuti e restituisca due indici m_1 ed m_2 con $1 \leq m_1 \leq m_2 \leq n$, in modo che $18 \leq V[i] \leq 22$ per ogni $1 \leq i \leq m_1$, $23 \leq V[i] \leq 26$ per ogni $m_1 + 1 \leq i \leq m_2$, $27 \leq V[i] \leq 30$ per ogni $m_2 + 1 \leq i \leq n$.

   L'algoritmo deve avere complessità di tempo $O(n)$ e deve funzionare *in loco*, cioè non utilizzare altro spazio di memoria oltre all'array V. Giustificarne la correttezza.

3. Descrivere ed analizzare un algoritmo di programmazione dinamica per la seguente variante al problema dello zaino. Dati n oggetti di peso $w_1, w_2, \cdots, w_n$ e valore $v_1, v_2, \cdots, v_n$ ed uno zaino di capacita' W, trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale e' al massimo W, con la condizione che un solo oggetto possa essere preso due volte (mentre tutti gli altri una sola volta). E' necessario commentare il funzionamento dell'algoritmo. L'esercizio potrà avere il massimo della valutazione solo se contiene anche lo pseudocodice e l'analisi dei costi computazionali.

4. Il grafo delle amicizie di Facebook è il grafo in cui i vertici sono gli utenti e vi è un arco fra ogni coppia di utenti amici. Descrivere ed analizzare un algoritmo che dato il grafo delle amicizie di Facebook, determini l'utente col numero massimo di amici e amici di amici, e restituisca tale numero massimo. Nota: attenzione ai duplicati.

   Esempio Se il grafo fosse il seguente, l'utente H avrebbe soltanto 2 fra amici ed amici di amici, mentre gli utenti D e G ne avrebbero 4, A ed E ne avrebbero 5, mentre gli utenti B, C, F avrebbero il numero massimo di amici ed amici di amici che è 6. L'algoritmo restituirebbe 6.



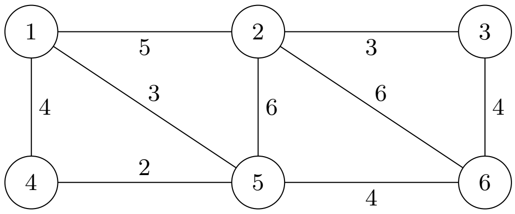
5. Si consideri il seguente grafo.

   a) Eseguire l'algoritmo di Kruskal per la ricerca del MST.

   b) Eseguire l'algoritmo *reverse-delete* per la ricerca del MST.

35 Appello 8 Aprile 2014

1. Si risolva la seguente relazione di ricorrenza, dove $T(1) = 1$:



$$T(n) = 3T(n/3) + n.$$

2. Descrivere **due** diversi algoritmi che verificano se un vettore $A[1..n]$ di interi e' una permutazione di $\{1, ..., n\}$. Giustificarne la correttezza. Confrontare la complessita' di tempo e di spazio dei due algoritmi proposti.

   Esempio. Per $n = 6$:

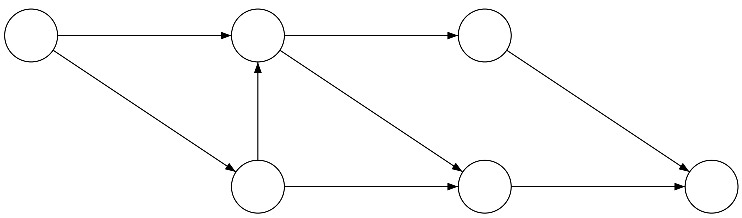
   se il vettore e' $[4, 2, 3, 5, 1, 6]$, l'algoritmo deve restituire Vero;

   se il vettore e' $[4, 2, 8, 5, 1, 6]$, l'algoritmo deve restituire Falso;

   se il vettore e' $[4, 2, 6, 2, 1, 6]$, l'algoritmo deve restituire Falso.

3. Descrivere ed analizzare un algoritmo di programmazione dinamica per la seguente variante al problema dello zaino. Dati n oggetti di peso $w_1, w_2, \cdots, w_n$ e valore $v_1, v_2, \cdots, v_n$ ed uno zaino di capacita' W, trovare il massimo valore di un sottoinsieme degli oggetti il cui peso totale e' al massimo W, con la condizione che un solo oggetto possa essere preso due volte (mentre tutti gli altri una sola volta). E' necessario commentare il funzionamento dell'algoritmo. L'esercizio potrà avere il massimo della valutazione solo se contiene anche lo pseudocodice e l'analisi dei costi computazionali.

4. Fornire un ordinamento topologico per il seguente DAG (*Directed Acyclic Graph*), utilizzando l'algoritmo studiato e spiegando le operazioni via via eseguite.



5. a) Dire quando un grafo $G = (V, E)$ e' bipartito.

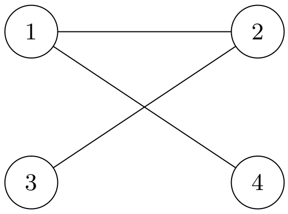
   b) Definire cos'e' un matching massimale in un grafo bipartito.

   c) Quanti distinti matching massimali vi sono nel seguente grafo biaprtito $G = (V, E)$?

   d) Trovare un matching massimale per il seguente grafo $G = (V, E)$ bipartito, eseguendo l'algoritmo studiato.

36 Appello 24 Giugno 2014

1. Si supponga di avere due algoritmi A ed A' che risolvono il medesimo problema in tempo $T_A(n)$ e $T_{A'}(n)$ rispettivamente. Se $T_A(n) = 4T(n/2) + n$ e $T_{A'}(n) = \sqrt{n^3}\log n + n$, quale dei



due algoritmi è asintoticamente più efficiente in termini di tempo? È necessario giustificare la risposta.

2. Si consideri il seguente algoritmo, dove A e' un array di n elementi i cui indici vanno da p ad r. Rispondere ai seguenti quesiti **giustificando** sempre la propria risposta.

```
Partition(A,p,r)
1. x = A[p]
2. i = p-1
3. j = r+1
4. while TRUE
5.    do  repeat j = j -1 until A[j] <= x
6.        repeat i = i +1 until A[i] >= x
7.        if i < j
8.              then scambia A[i] <-> A[j]
9.               else return j
```

a) Qual e' l'effetto dell'esecuzione di Partition sull'array A?

b) Cosa succede se le linee 5 e 6 vengono sostituite con queste?

```
5.    do  repeat j = j -1 until A[j] < x
6.        repeat i = i +1 until A[i] > x
```

c) Cosa succede se le linee 5 e 6 vengono sostituite con queste?

```
5.    do  repeat j = j -1 until A[j] >= x
6.        repeat i = i +1 until A[i] <= x
```

3. Si supponga di avere un listello di legno che vogliamo tagliare in alcune specifiche posizioni. Il falegname si fa pagare x euro per tagliare un listello lungo x centimetri in due pezzi, in qualsiasi punto si faccia il taglio. Il problema consiste nel trovare il costo minimo totale per tagliare un listello lungo L nei punti P_1, P_2, $\cdots$, P_n, le cui distanze da una fissata estremita' siano rispettivamente x_1, x_2, $\cdots$, x_n.

   Esempio. Supponiamo di avere un listello lungo $L = 10$ cm e volere effettuare dei tagli nei punti P_1, P_2, P_3, P_4, distanti da una estremita' fissata, rispettivamente: $x_1 = 3$, $x_2 = 6$, $x_3 = 7$, $x_4 = 9$. Se effettuo i tagli nell'ordine nei punti P_1, P_2, P_3, P_4, il costo sara': $10+7+4+3 = 24$ euro. Se effettuo i tagli nell'ordine nei punti P_4, P_1, P_3, P_2, il costo sara': $10+9+6+4 = 29$ euro.

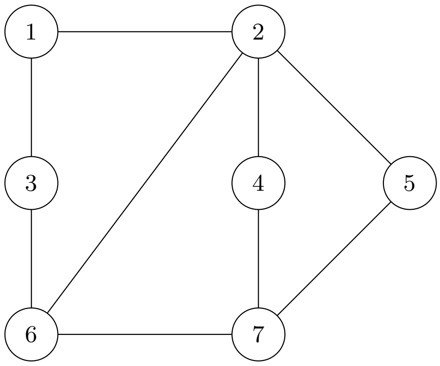
   Si considerino gli algoritmi greedy basati rispettivamente sui due seguenti criteri di scelta per effettuare il prossimo taglio:

   a) scelgo i punti secondo le distanze x_i crescenti

b) scelgo il punto piu' vicino alla meta' del listello (e in caso vi fossero due punti equidistanti dalla meta' scelgo quello con distanza x_i minore)

Per ognuno dei criteri dire se l'algoritmo corrispondente porta sempre ad una soluzione ottimale oppure no, **giustificando** la risposta.

4. Sia G il grafo seguente:



a) Eseguire la BFS su G a partire dal nodo 1, mostrando i "layers" (strati) ottenuti e l'albero BFS risultante.

b) Dire se G e' un grafo bipartito, motivando la risposta e, in caso affermativo, fornire la partizione dei vertici per cui G e' un grafo bipartito.

c) Trovare un matching massimale in G eseguendo l'algoritmo studiato.

5. a) Definire cos'è un DAG.

b) Definire cos'è un ordinamento topologico in un grafo orientato.

c) Descrivere un algoritmo che dato un grafo orientato G, decida se G è un DAG oppure no. Giustificarne la correttezza.

d) Eseguire l'algoritmo descritto al punto c) sul seguente grafo.