Progettazione di Algoritmi (9 CFU)
11/6/2019

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

1	2	3	4	5	6	Totale
/20	/20	/15	/15	/15	/15	/100

1. Analisi degli algoritmi e notazione asintotica

a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.
   1. $n^2 \log^2 n+100n^2 = O(n^4)$
   2. $n^{1/8} = \Omega(n^{1/4})$
   3. $n^{1/3} = O(\log n)$
   4. $n^3-1000n^2+8=\Omega(n^3)$
   5. $\log(\log n)=O((\log n)(\log n))$

b) Si dimostri che se $0<f(n)=O(h(n))$ e $1<g(n)= \Omega(p(n))$ allora $f(n)/g(n)= O(h(n)/p(n))$. Occorre utilizzare solo la definizione di O e di Ω e nessuna altra proprieta`.

c) Si analizzi il tempo di esecuzione nel caso pessimo del seguente segmento di codice fornendo una stima asintotica **quanto migliore e` possibile** per esso. **Si giustifichi in modo chiaro la risposta.**

```
FOR(i=1; i<n; i=5*i){
  FOR(j=0; j<10; j=j+1) {
      print(j);
   }
}
```

2. Divide et Impera

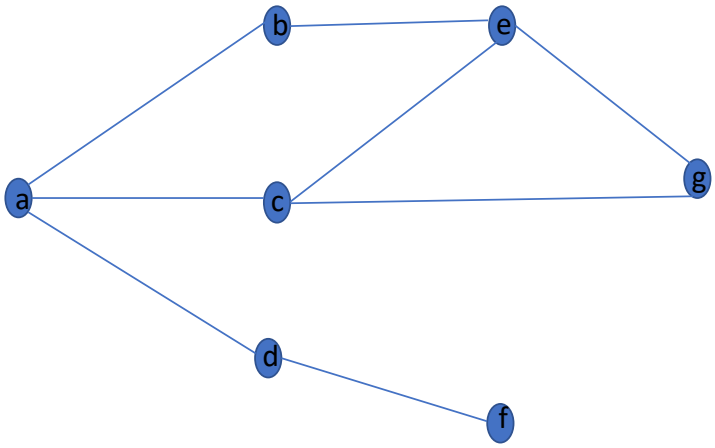
a) Si scriva lo pseudocodice di un algoritmo ricorsivo che prende in input un array di numeri e computa il massimo numero di occorrenze consecutive del numero '2'. Ad esempio, se l'array contiene la sequenza <2 2 3 6 2 2 2 2 3 3> allora l'algoritmo restituisce 4. Per ogni procedura, occorre specificare l'input e l'output della procedura.

b) Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo al punto a) e si fornisca una stima quanto migliore e` possibile del tempo di esecuzione dell'algoritmo a partire dalla suddetta relazione di ricorrenza.

3. Grafi

a) Si scriva lo pseudocodice dell'algoritmo che effettua la visita BFS di un grafo utilizzando una coda FIFO e costruisce l'**albero BFS**. Si analizzi il tempo di esecuzione dell'algoritmo proposto. Analizzare il tempo di esecuzione significa fornire un limite superiore asintotico quanto migliore e` possibile al tempo di esecuzione dell'algoritmo **giustificando la risposta**.

b) Si disegni l'albero DFS generato da una visita DFS del seguente grafo a partire dal nodo sorgente **a**. Si assuma che i nodi siano disposti nelle liste di adiacenza in base all'ordine crescente delle proprie etichette.

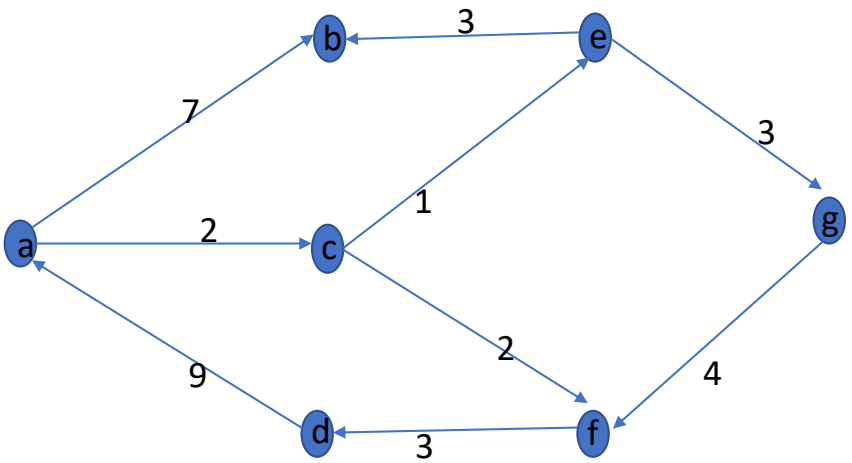


4. **Algoritmi greedy**

a) Si spieghi in che cosa consiste un'istanza (input) del problema del caching offline e in cosa consiste una soluzione (output) del problema. Se dalla risposta a questo punto si evincera` che lo studente non sa in cosa consiste il problema del caching offline, i punti successivi dell'esercizio non saranno valutati.

b) Si spieghi che cosa e` un eviction scheduling ridotto (se non lo si e` gia spiegato al punto precedente) e si dimostri che e` sempre possibile trasformare un eviction schedule in un eviction schedule ridotto senza aumentare il numero totale di inserimenti nella cache.

c) Si mostri l'esecuzione dell'algoritmo di Dijkstra sul seguente grafo a partire dal nodo sorgente a**. Per ogni passo occorre mostrare l'albero dei percorsi minimi costruito fino a quel momento e il contenuto della coda a priorita` (indicando anche le chiavi). Nel caso in cui non vengano mostrati tutti i passi, il punteggio sara` di 0 punti.**



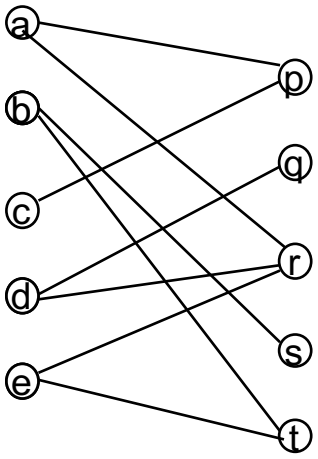
Progettazione di Algoritmi (9 CFU)
11/6/2019

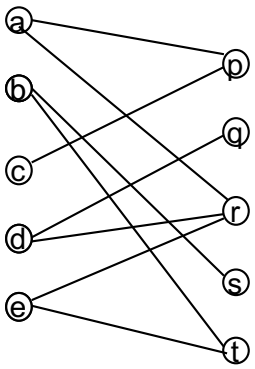
5. **Programmazione dinamica**

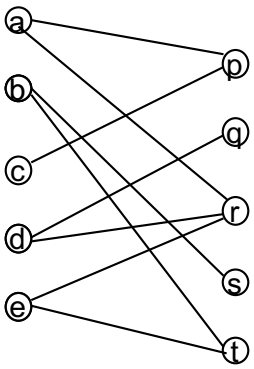
a) Si descriva in modo chiaro e schematico in che cosa consiste un'istanza del problema dell'interval scheduling pesato e qual e` l'obiettivo del problema. Se dalla risposta a questo punto si evincera` che lo studente non sa in cosa consiste il problema dell'interval scheduling pesato, i punti successivi dell'esercizio non saranno valutati.

b) Fornire una formula per il calcolo del valore della soluzione ottima per il problema dell'interval scheduling pesato in termini di valori delle soluzioni ottime per sottoproblemi di taglia piu` piccola. **Spiegare in modo chiaro e schematico come si arriva alla formula da voi fornita definendo in modo chiaro tutte le quantita` e parametri che compaiono nella formula. In assenza di queste spiegazioni l'esercizio sara` valutato 0 punti.**

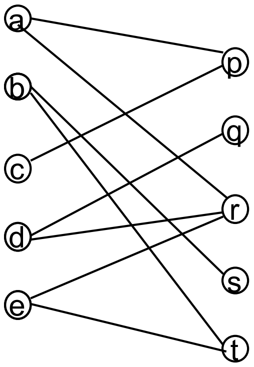
6. **Massimo flusso**
   a) Mostrare i passi dell'algoritmo per trovare il massimo matching di un grafo bipartito sul seguente grafo. **Il primo passo deve "utilizzare" l'arco (a,p). Se il primo passo non effettua la scelta indicata l'esercizio verra` valutato 0 punti.** Occorre mostrare l'esecuzione di **tutto** l'algoritmo e mostrare il matching prodotto.
   Per vostra comodita`, di seguito sono riportate diverse coppie di immagini del grafo. Il numero di coppie non e` indicativo del numero di iterazioni fatte dall'algoritmo.

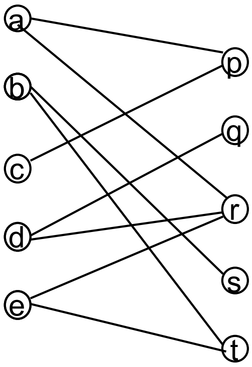


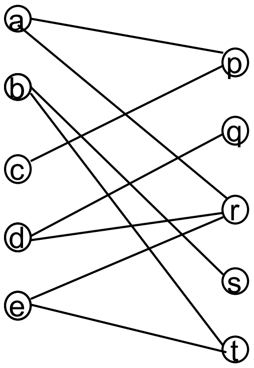


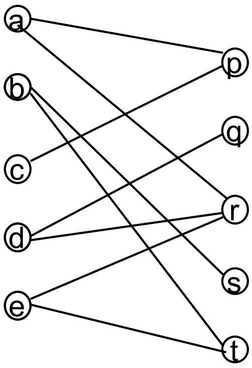


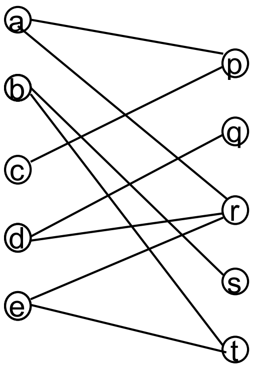
Progettazione di Algoritmi (9 CFU)
11/6/2019

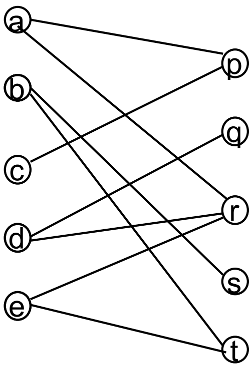




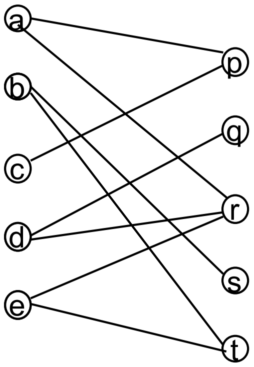


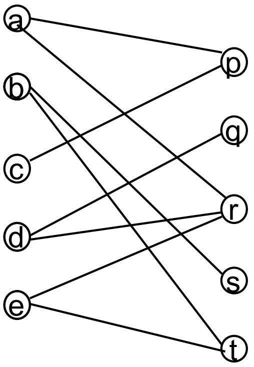


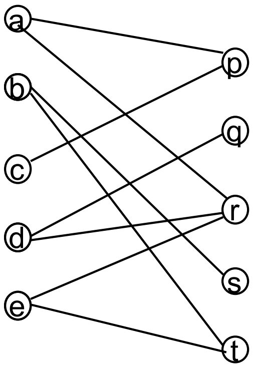


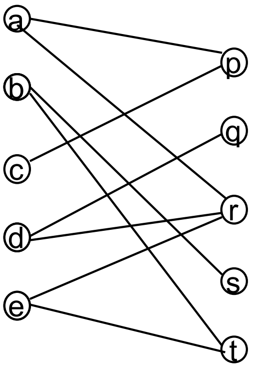


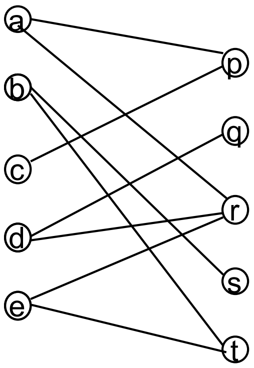
Progettazione di Algoritmi (9 CFU)
11/6/2019

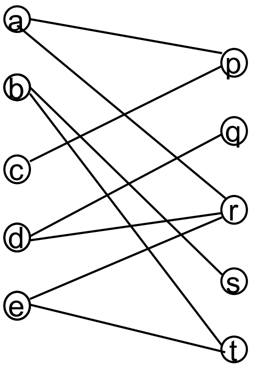












b) Descrivere in modo chiaro il comportamento dell'algoritmo che computa il massimo numero di percorsi disgiunti tra due nodi di un grafo direzionato. Occorre anche descrivere in che modo l'algoritmo alla fine genera i percorsi disgiunti.