

**si prenda visione anche del file con gli esercizi aggiuntivi**

**9 cfu nuovo**

1	2	3	4	5	Totale
/18	/20	/22	/18	22/	/100

**9 cfu vecchio**

1	2	3	4	5	6	Totale
/18	/20	/22	/10	17/	/13	/100

**6 cfu nuovo**

3	4	5	Totale
/40	/28	/32	/100

**6 cfu vecchio**

3	4	5	6	Totale
/33	/22	/20	/25	/100

**Si ricorda che per i punti che richiedono l'analisi di un algoritmo occorre fornire un limite superiore asintotico quanto migliore e' possibile al tempo di esecuzione dell'algoritmo giustificando la risposta.**

**1. Analisi degli algoritmi e notazione asintotica**

- a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.
1.  $n^4 2^n = O(4^n)$
  2.  $n^n = \Omega(n^{3n/2} n^{n/2})$
  3.  $n(\log n) = \Theta(\log n^{2n})$
  4.  $n^3 + 1000n^2 + 100 = \Theta(n^3 + n)$
  5.  $\log(\log n) + n = O((\log n)^{1/2} + n^{1/2})$
- b) Si dimostri che se  $0 < f(n) = O(h(n))$  allora  $a(f(n)) = O(h(n))$ , dove  $a$  è una costante positiva. Occorre utilizzare solo la definizione di  $O$  e nessuna altra proprietà.
- 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.**

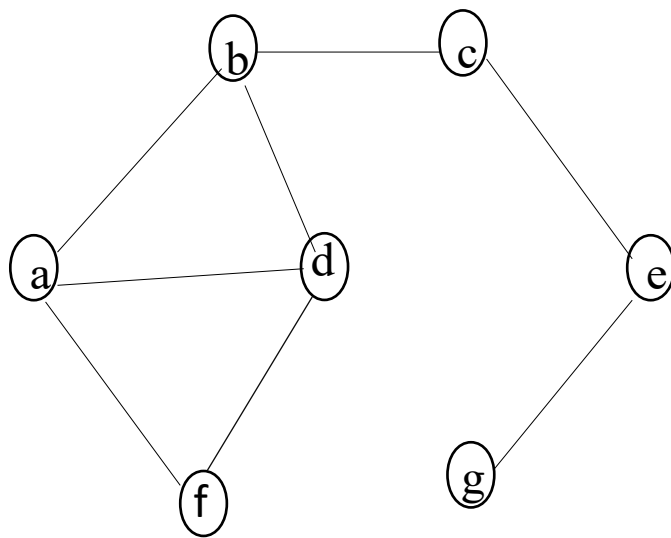
```
FOR(i=1; i<n; i=i+1){
  FOR(j=1; j<3^i; j=j*3) {
    print(j);
  }
}
```

## 2. Divide et Impera

- a) Si scriva lo pseudocodice di un algoritmo ricorsivo **conta(u,x)**, dove **u** è un nodo di un albero binario e **x** un elemento. L'algoritmo restituisce il numero di nodi che contengono **x** nell'albero avente come radice **u**. (L'elemento di un nodo **x** è contenuto in **x.dato**).
- b) Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo QuickSort nel caso pessimo .
- c) A partire dalla relazione di ricorrenza da voi fornita al punto b), si fornisca una funzione  $h(n)$  tale  $T(n)=O(h(n))$ . Giustificare la risposta usando il metodo iterativo.

### 3. Grafi

- a) Si scriva lo pseudocodice **dell'algoritmo di Dijkstra** che fa uso di una **coda a priorit ** aggiungendo anche le linee di codice per la costruzione dell'albero dei cammini minimi. Si analizzi il tempo di esecuzione dell'algoritmo proposto quando la coda a priorit    implementata con un heap binario.
- b) Si disegni l'albero DFS generato da una visita DFS del seguente grafo a partire dal nodo sorgente **a** numerando i nodi dell'albero con interi consecutivi che indicano in che ordine i nodi sono stati visitati. Si assuma che i nodi siano disposti nelle liste di adiacenza in base all'ordine crescente delle proprie etichette.



- c) Si descriva un algoritmo che prende in input un grafo non direzionato e, nel caso in cui il grafo contenga un ciclo, restituisce una lista contenente i nodi che formano il ciclo e in caso contrario restituisce una lista vuota. L'algoritmo deve avere tempo di esecuzione  $O(n+m)$  nel caso pessimo.

#### 4. Algoritmi greedy

- a) Si spieghi in che cosa consiste un'istanza (input) del problema della minimizzazione dei ritardi e qual è l'obiettivo del problema (output). Devono essere definite anche tutte le quantità utilizzate per spiegare l'obiettivo del problema. Se dalla risposta a questo punto si evincerà che lo studente non sa in cosa consiste il problema della minimizzazione dei ritardi, i punti successivi dell'esercizio non saranno valutati.
- b) Si fornisca un'istanza del problema della minimizzazione dei ritardi con  $n=6$  per cui il valore della soluzione ottima è 5 e al più due attività hanno ritardo 0 nella soluzione ottima. Si mostri chiaramente perché il valore della soluzione per l'istanza da voi fornita è 5 e qual è il ritardo di tutte le attività.
- c) [6 cfu tutti e 9 cfu nuovo] Si scriva lo pseudocodice dell'algoritmo greedy che restituisce la soluzione ottima per il problema della minimizzazione dei ritardi.

2) ci viene dato in input ~~sa~~ i lavori da fare che il ritardo massimo entro i quali devono essere eseguiti il tempo di fine è il tempo finale totale

lavoro che richiede tempo 3 da finire entro tempo 6  
// 2 //

// 8

//

5. **Programmazione dinamica**

- a) [9 cfu nuovo e 6 cfu nuovo] Fornire una formula per il calcolo del valore della soluzione ottima del problema che consiste nel trovare il valore massimo ottenibile con una parentesizzazione completa di un'espressione contenente solo l'operatore della divisione. Spiegare **in modo chiaro** quanto segue:
1. cosa rappresentano la funzione MAX, i suoi parametri ed eventuali altre funzioni da voi utilizzate nella formula
  2. come si arriva alla formula fornita per MAX e a quelle di eventuali altre funzioni utilizzate per scrivere la suddetta formula.
- b) Scrivere l'**algoritmo ricorsivo** di programmazione dinamica per **il problema dello zaino**.
- c) Si consideri la seguente istanza del problema dello **zaino**.  $n=4$ ;  $w_1=2$ ,  $w_2=2$ ,  $w_3=5$ ,  $w_4=1$ ;  $v_1=2$ ,  $v_2=4$ ,  $v_3=8$ ,  $v_4=9$ ; soglia:  $W=6$  (dove i  $w_i$  sono i pesi e  $v_i$  i valori). Si disegni la tabella M dei valori  $M[i,w]$  computati dall'algoritmo per il problema dello zaino basato sulla programmazione dinamica. Dire qual è la soluzione ottima e qual è il suo valore. Indicare con un cerchio le celle ispezionate dall'algoritmo che costruisce la soluzione ottima.