

Inviare l'elaborato a algoritmi2020@gmail.com

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	/32	/28	/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

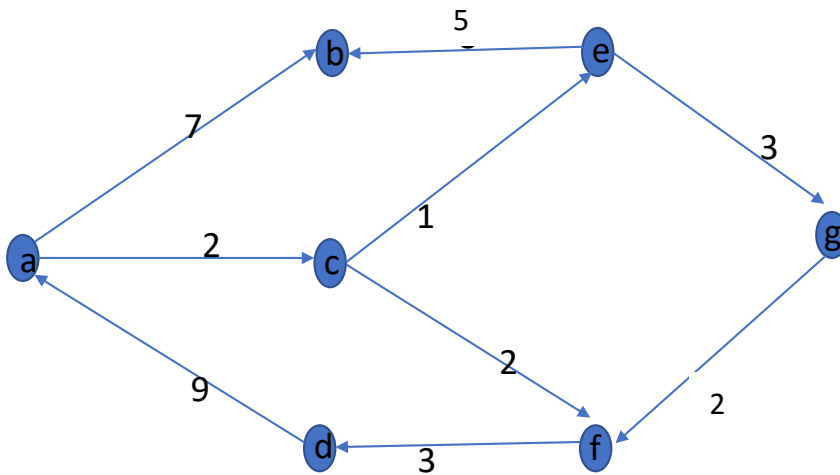
- a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.
1. $4n^4 = O(n^3)$
 2. $n^{\log n} = \Omega((\log n)^n)$
 3. $n(\log n)^{1/2} + n^{1/3} = O(\log n^n)$
 4. $n^3 + 10 = \Omega(n^3 + 10n)$
 5. $2^{\log_4 n} = \Theta(n)$
- b) Si dimostri che se $1 < f(n) = O(h(n))$ allora $(f(n))^a = O(h(n)^a)$, 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.**
- ```
i=1
j=1
while(i<n and j<m)
 if (i <= j)
 i=i+1
 else
 j=j+1
```

## 2. Divide et Impera

- [9 cfu nuovo] Si descriva il comportamento dell'algoritmo per determinare la coppia di punti più vicina in un insieme di punti del piano.
- Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo QuickSelect nel caso pessimo. **Si giustifichi in modo chiaro la risposta.**
- 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

- Si scriva lo pseudocodice **dell'algoritmo BFS** che fa uso di una **coda FIFO** aggiungendo anche le linee di codice per la costruzione dell'albero BFS. Si analizzi il tempo di esecuzione dell'algoritmo proposto.
- Si mostri l'esecuzione dell'algoritmo di Dijkstra sul seguente grafo a partire dal nodo sorgente **a**. **Per ogni passo si mostri il contenuto della coda a priorit , incluse le chiavi degli elementi, e l'albero dei percorsi minimi costruito fino a quel passo.**



- Si consideri l'algoritmo che determina se un grafo   bipartito. Si dimostri che se l'algoritmo colora due nodi adiacenti dello stesso colore allora il grafo non   bipartito.

#### 4. Algoritmi greedy

- a) Si spieghi in che cosa consiste un'istanza (input) del problema del partizionamento di intervalli e qual è l'obiettivo del problema (output). Se dalla risposta a questo punto si evincerà che lo studente non sa in cosa consiste il problema, i punti successivi dell'esercizio non saranno valutati.
- b) Si scriva lo pseudocodice dell'algoritmo greedy che restituisce il valore della soluzione ottima per il problema del partizionamento di intervalli. SI SCRIVA LO PSEUDOCODICE IN ITALIANO: le uniche parole inglesi consentite sono le parole chiave if, for, ecc. .
- c) [6 cfu tutti e 9 cfu nuovo] Si fornisca un'istanza del problema **dell'Interval Scheduling** per la quale la strategia greedy "Fewest Conflicts" non fornisce una soluzione ottima. Occorre indicare i tempi di inizio e di fine di ogni job.

#### 5. Programmazione dinamica

- a) Scrivere lo pseudocodice di un algoritmo basato sul principio di programmazione dinamica per computare la sottosequenza comune più lunga di due sequenze.
- b) Fornire una formula per il calcolo del valore della soluzione ottima OPT del problema dei cammini minimi in termini di valori delle soluzioni ottime per sottoproblemi di taglia più piccola. Spiegare in modo chiaro
  1. cosa rappresenta la funzione OPT e cosa rappresentando i suoi parametri
  2. come si arriva alla formula da voi fornita.
- c) Si consideri il seguente grafo e **si disegni la matrice 3 x 3** dei valori computati dall'algoritmo di Bellman-Ford per il problema dei cammini minimi quando il nodo t ricevuto in input dell'algoritmo è il nodo a.

