

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

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

1. Analisi degli algoritmi e notazione asintotica (solo per quelli da 9 cfu)

a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.

$$n^{1/10} + n \log n = O(n \log n + n^{1/13})$$

$$n^{1/4} + n^5 = O(n^4 \log n^{1/2})$$

$$n^{\log n} = O((\log n)^n)$$

$$n^2 - n = \Theta(n^2 + n)$$

$$n^{100} = O(100^n)$$

b) Si dimostri che se $1 < f(n) = O(h(n))$ e se $1 < g(n) = O(q(n))$ allora $af(n) + g(n) = O(h(n)q(n))$, dove a è una costante positiva. Occorre utilizzare solo la definizione di O e nessuna altra proprietà.

b) Si calcolino le costanti n_0 e c per le quali si può dire che $2n^2 + 100n^{1/2} = O(n^2)$. Si mostrino tutti i passaggi dei calcoli giustificandoli.

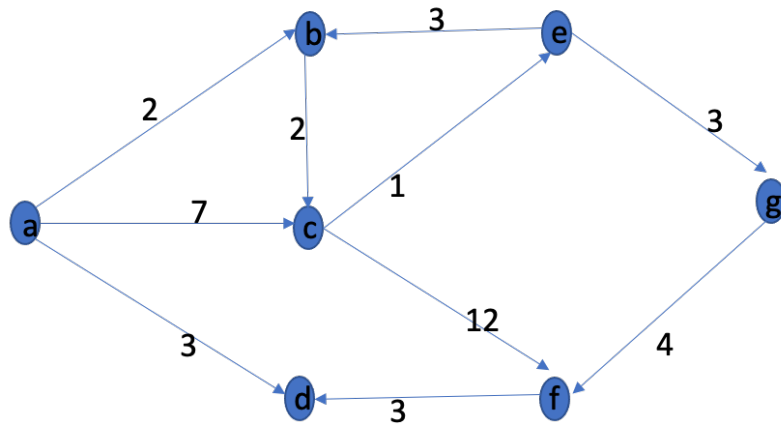
- c) Si analizzi il tempo di esecuzione nel caso pessimo del seguente segmento di codice fornendo una stima asintotica **quanto migliore è possibile** per esso. **Si giustifichi in modo chiaro la risposta.**

```
i=0;
j=1;
while(i<=n or j<=m){
    i=i+1
    j=j*3
}
```

2. Grafi

- a) Si scriva lo pseudocodice **dell'algoritmo ricorsivo DFS** aggiungendo anche le linee di codice per la costruzione dell'albero DFS. Si analizzi il tempo di esecuzione dell'algoritmo proposto.

- b) Si consideri l'algoritmo topologico modificato nel seguente modo: se il grafo è un DAG, l'algoritmo restituisce l'ordinamento topologico del grafo; se il grafo non è un DAG, l'algoritmo restituisce un ciclo direzionato del grafo. Si mostri l'esecuzione di questo algoritmo sul seguente grafo. Per ogni chiamata ricorsiva effettuata, occorre mostrare il grafo su cui essa è invocata. Se il grafo è un DAG alla fine occorre fornire l'ordinamento topologico mentre se il grafo non è una DAG occorre fornire un ciclo del grafo e spiegare come questo ciclo viene individuato dall'algoritmo (**per il particolare grafo considerato non in generale**).



- c) Si consideri l'algoritmo di Prim.
- In che punto dell'algoritmo viene effettuata la scelta greedy
 - Cosa rappresenta la chiave assegnata a un nodo nella coda a priorit 
 - Per quale motivo la scelta greedy assicura che venga selezionato un arco che fa parte dello MST.
 - Qual   il costo dell'algoritmo quando la coda   implementata con un heap binario?
Si computi il costo della fase di inizializzazione, il costo delle operazioni che usano la coda a priorit  e il numero totale di volte in cui ciascuna di queste viene eseguita giustificando la risposta in modo chiaro.

2. Greedy

- a) Si fornisca un'istanza di minimizzazione dei ritardi formata da 7 attività con almeno 4 diverse soluzioni ottime. La soluzione ottima (in questo caso le soluzioni ottime) deve avere valore 11 e non più di una attività in essa può avere ritardo 0. **Si indichino chiaramente i valori numerici dell'input e si illustri graficamente come vengono schedate le attività. Si calcolino tutti i ritardi.**

b) Si consideri il seguente problema:

Abbiamo un asse di legno dal quale dobbiamo cercare di ricavare n pezzi aventi lunghezze l_1, l_2, \dots, l_n . Il nostro obiettivo è di ricavare quanti più segmenti è possibile.

- i. Descrivere una strategia greedy che permette di ottenere il massimo numero di segmenti.
- ii. Dimostrare che per ogni $j \geq 0$, un algoritmo ottimo che fa le stesse prime j scelte di greedy può essere trasformato in un algoritmo ottimo che fa le stesse $j+1$ scelte di greedy.

- c) Si consideri il problema del partizionamento di intervalli.

Si scriva lo pseudocodice dell'algoritmo che restituisce l'insieme delle coppie (j,v) , dove j è un'attività e v è la risorsa utilizzata per eseguire v . Lo pseudocodice deve essere scritto in italiano, eccezion fatta per le parole chiave, quali "if", "else", "while", "for", ecc. **Occorre specificare bene l'input e l'output.**

3. Divide et Impera

a)

- I. Si scriva lo pseudocodice di un algoritmo di divide et impera che restituisce l'elemento di rango 10 di un array (decimo elemento in ordine di grandezza). L'algoritmo NON deve prima ordinare l'array.
- II. Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo da voi fornito al punto I **nel caso pessimo giustificando la risposta**.
- III. A partire dalla relazione di ricorrenza da voi fornita al punto b), si fornisca una funzione $h(n)$ tale che $T(n)=O(h(n))$. **Giustificare la risposta** usando o il metodo iterativo o quello della sostituzione (induzione).

- b) Supponiamo di avere un array di numeri A ordinato in modo non decrescente e che ciascun elemento appaia al più 8 volte in A. Scrivere un algoritmo di divide et impera, quanto più efficiente è possibile, che restituisce una lista contenente tutti gli indici delle celle in cui compare un dato elemento x nell'array A.
Dire qual è il tempo di esecuzione del vostro algoritmo nel caso pessimo (non occorre giustificare l'analisi).

4. Programmazione dinamica.

- a) Si consideri il problema dell'interval scheduling pesato.
 - i. Si spieghi cosa rappresenta OPT e cosa rappresenta il suo parametro.
 - ii. Si fornisca una formula di ricorrenza per il calcolo di OPT definendo in modo chiaro **tutte le quantita'** utilizzate nella formula.
 - iii. Si spieghi in modo chiaro come si arriva alla formula da voi fornita

- b) Scrivere lo pseudocodice di un algoritmo **iterativo di programmazione dinamica** che calcola il valore della soluzione ottima per il problema di **Subset Sums** modificato in modo tale che, oltre ai valori della tabella M, vengano computati i valori di una tabella Q che puo' essere usata per stampare la soluzione ottima. In altre parole, deve essere possibile scrivere un algoritmo che riceve in input **solo** Q e stampa la soluzione ottima (non occorre anche scrivere l'algoritmo che stampa la soluzione ottima).

- c) Si fornisca un'istanza del problema della sottosequenza comune piu' lunga che consista di due stringhe di lunghezza 6 che hanno come sottosequenza comune piu' lunga una sequenza di lunghezza 3. Si fornisca la tabella M computata dall'algoritmo che calcola il valore della soluzione ottima per la vostra istanza. Una volta disegnata la tabella M, si contrassegnino con un cerchio le entrate $M[i,j]$ corrispondenti alle coppie di indici (i,j) su cui viene invocato ricorsivamente l'algoritmo che ricostruisce la soluzione ottima e si mostri la soluzione ottima.