

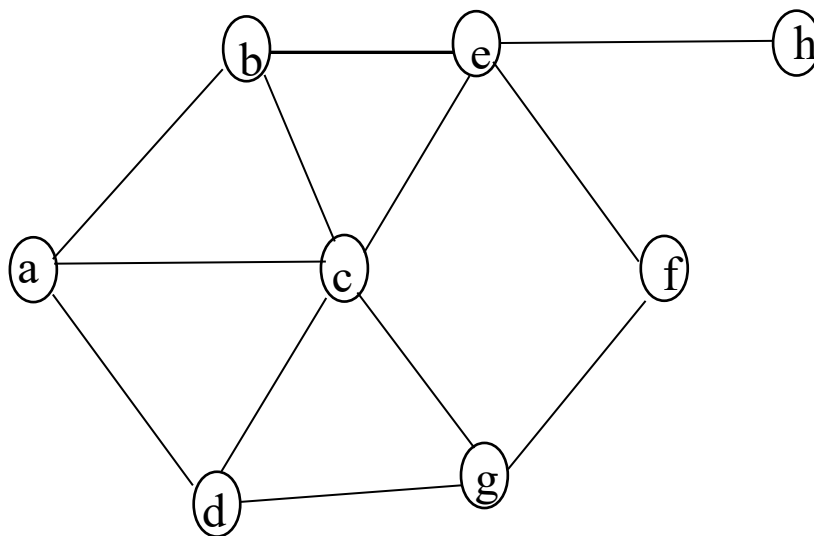
Cognome e Nome:  
Numero di Matricola:

**1. Grafi [20 minuti] (per quelli da 6 continua alla fine)**

a) MST

- i. Scrivere lo pseudocodice dell'algoritmo di Prim con le linee per costruire il minimo albero ricoprente.
- ii. Spiegare a parole cosa rappresenta la chiave associata ad un nodo nella coda a priorit .
- iii. Spiegare perche' l'arco inserito ad un generico passo dall'algoritmo fa sicuramente parte del minimo albero ricoprente.

- 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.



- c) Si mostri l'esecuzione dell'algoritmo per l'ordinamento topologico sul DAG formato dai seguenti archi (a,b),(a,f),(b,c) (b,e)(b,d) (c,d) (e,c) (e,f). Si disegnino i grafi su cui vengono invocate le chiamate ricorsive e alla fine si fornisca l'ordinamento topologico del grafo.

2. **Algoritmi greedy [20 minuti] (per quelli da 6 cfu continua alla fine)**

- a) Fornire un'istanza del partizionamento di intervalli di dimensione  $n=9$  e con valore della soluzione ottima uguale a 4. Indicare inoltre in quali istanti l'algoritmo greedy che trova la soluzione ottima incrementa il valore della soluzione **per la vostra istanza. Non è sufficiente fornire un disegno che descriva l'input.**
- b) Si fornisca un'istanza del problema della minimizzazione dei ritardi con  $n=5$  per cui il valore della soluzione ottima è 8 e al più una attività ha ritardo 0 nella soluzione ottima. Si mostri chiaramente perché il valore della soluzione per l'istanza da voi fornita è 8 e qual è il ritardo di tutte le attività.
- c) Si consideri il problema della minimizzazione dei ritardi. Si dimostri che in scheduling senza inversioni e senza idle time, job con la stessa scadenza vengono eseguiti uno dopo l'altro.

3. **Programmazione dinamica [20 minuti] (per quelli da 6 cfu continua alla fine)**

- a) Fornire una formula per il calcolo del valore della soluzione ottima OPT del problema dello zaino 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 rappresentano i suoi parametri
  2. come si arriva alla formula da voi fornita.
- b) Si fornisca la tabella M costruita dall'algoritmo che computa il valore della soluzione ottima per **subset sums** quando l'istanza input è  $w_1=2, w_2=3, w_3=5, w_4=5, w_5=3, W=6$ . Una volta costruita la tabella M, si contrassegnino con un cerchio le entrate  $M[i,w]$  corrispondenti alle coppie di indici  $(i,w)$  su cui viene invocato ricorsivamente l'algoritmo che ricostruisce la soluzione ottima.
- c) **Si fornisca** un controesempio di 4 job che dimostri che la strategia greedy che permette di ottenere la soluzione ottima per il problema dell'Interval Scheduling non sempre fornisce la soluzione ottima per il problema dell'interval scheduling pesato. **Si fornisca** poi un'istanza di interval scheduling pesato per la quale invece la suddetta strategia greedy ottiene la soluzione ottima. **Si giustificino le risposte.**

4. **Divide et Impera [20 minuti]**

- a) Si scriva lo pseudocodice dell'algoritmo **trova**, basato sul **divide et impera**, che restituisce **true** se  $x$  è presente nell'array  $A$  e **false** se  $x$  non è presente. L'elemento  $x$  e l'array  $A$  vengono passati in input all'algoritmo insieme ad eventuali altri argomenti. Non è detto che gli elementi dell'array  $A$  siano ordinati.  
**NB: un algoritmo che richiama ricorsivamente se stesso su un segmento contenente solo un numero costante di elementi in meno rispetto al segmento input (ad esempio, un elemento in meno) non è propriamente un algoritmo di divide et impera e non sarà considerato corretto.**
- b) Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo da voi fornito al punto a) **giustificando la risposta**.
- c) 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).

**5. Analisi degli algoritmi e notazione asintotica [25 minuti]**

- a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.
1.  $n + (\log n)^4 = O(n + \log(n^3))$
  2.  $\log(\log n) = O((\log n)^{1/4})$
  3.  $n^n = \Theta(n^{n/4})$
  4.  $n^{1/2} = O(\log n)$
  5.  $n^{3/2} + n^2 = \Theta(n^2)$
- b) Dimostrare che la seguente affermazione e' vera giustificando la risposta. Occorre fornire le costanti  $c$  ed  $n_0$ .
1.  $n^2 + 3n = O(n^2)$
- c) Dimostrare che la seguente affermazione e' falsa giustificando **matematicamente** la risposta.
1.  $n^3 = \Omega(n^4)$
- d) Si dimostri che se  $0 < f(n) = O(h(n))$  e  $0 < g(n) = O(p(n))$  e **a** e **b** sono costanti positive allora  **$af(n) + bg(n) = O(h(n) + p(n))$** . Occorre utilizzare solo la definizione di  $O$  e nessuna altra proprieta'.
- e) 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.**

```
FOR(i=1; i<n; i=3*i){  
    FOR(j=1; j<100; j=j*2) {  
        print(j);  
    }  
}
```

Esercizi aggiuntivi per 6 cfu [25 minuti]

1

- d) Si mostri l'esecuzione dell'algoritmo di Prim sul grafo contenente i seguenti archi (a,b)**4**, (a,c)**3**, (a,d)**7**, (b,e)**2**, (c,b)**5**, (c,d)**1**, (c,f)**2**, (d,f)**6**. I numeri rossi sono i pesi degli archi. La radice dell'albero deve essere il nodo a. **Per ogni passo occorre mostrare l'albero costruito fino a quel momento e il contenuto della coda a priorit  (indicando anche le chiavi).**

2

- d) Si scriva lo pseudocodice dell'algoritmo greedy ottimo per l'interval scheduling e si analizzi il tempo di esecuzione nel caso pessimo **giustificando la risposta.**

3

- d) Si scriva lo pseudocodice dell'algoritmo di Bellman-Ford. **Analizzare il tempo di esecuzione giustificando nel dettaglio la risposta (non serve dare solo il tempo di esecuzione).**