

Cognome e Nome:  
Numero di Matricola:

**Spazio riservato alla correzione**

1	2	3	4	5	Totale
/20	/19	/21	/22	18	/100

per i compiti da 6 CFU:

/35      /30      /35

La prova per l'esame di 6 CFU consiste degli esercizi 1, 2, 3 e di quelli riportati nell'ultima pagina (che non fanno parte della prova dell'esame da 9 CFU).

**1. Grafi**

- a) Si scriva lo pseudocodice dell'algoritmo **ricorsivo** che effettua la visita DFS di un grafo e costruisce l'**albero DFS**. Si analizzi il tempo di esecuzione dell'algoritmo proposto. Analizzare il tempo di esecuzione significa fornire un limite superiore asintotico quanto migliore è possibile al tempo di esecuzione dell'algoritmo **giustificando la risposta**.
- b) Si disegni l'albero BFS generato da una visita BFS del grafo contenente i seguenti archi (a,b),(a,c), (a,d), (b,e), (b,c), (c,f),(c,g),(d,f), (e,f),(e,g),(f,g). La visita deve partire dal nodo sorgente **a**. Riportare vicino a ciascun nodo dell'albero un intero che indichi quando il nodo viene scoperto. Si assuma che i nodi siano disposti nelle liste di adiacenza in base all'ordine crescente delle proprie etichette.
- c) Si indichino gli archi che devono essere rimossi dal grafo dell'esercizio c) affinché nel grafo non vi siano più cicli. **Per individuare gli archi da cancellare dovete utilizzare una tecnica algoritmica che possa essere applicata mediante ad un arbitrario grafo. Spiegare la tecnica che avete utilizzato per cancellare i suddetti archi. L'esercizio sarà valutato 0 punti se saranno semplicemente indicati gli archi da cancellare e i cicli su quali essi si trovano.**

## 2. Algoritmi greedy [20 minuti]

- a) Si scriva lo pseudocodice dell'algoritmo che restituisce la soluzione ottima per Interval Scheduling e si fornisca una stima asintotica del tempo di esecuzione dell'algoritmo nel caso pessimo giustificando con chiarezza la risposta.
- b) Si consideri il problema dell'interval scheduling e sia  $i_1, i_2, \dots, i_k$  l'insieme di job selezionati dall'algoritmo greedy e sia  $j_1, j_2, \dots, j_m$  l'insieme di job nella soluzione ottima. Entrambe le sequenze di job  $i_1, i_2, \dots, i_k$  e  $j_1, j_2, \dots, j_m$  sono ordinate rispetto ai tempi di fine. Supponiamo che qualcuno ci abbia detto che l'esecuzione dei job  $i_1, i_2, \dots, i_k$  termina non più tardi di quella dei job  $j_1, j_2, \dots, j_k$ . **A partire da questa informazione, dimostrate che la soluzione greedy è ottima.**
- c) Si mostri l'esecuzione dell'algoritmo di Prim sul grafo contenente i seguenti archi (a,b)6, (a,c)3, (a,d)5, (a,e)8, (b,e)2, (c,d)3, (c,e)4, (c,f)2, (d,f)4. 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).**

3. **Programmazione dinamica [25 minuti]**

- a) Si consideri il problema dei cammini minimi.
  - I. Si spieghi in modo chiaro cosa rappresenta  $OPT(i,v)$  e cosa rappresentano i parametri  $i$  e  $v$ . Nel fare questo, spiegare in modo chiaro **anche** cosa si intende per soluzione ottima.
  - II. Si fornisca una relazione di ricorrenza per il calcolo di  $OPT(i,v)$  spiegando in modo chiaro come si ottiene la formula.
  
- b) Si scriva lo pseudocodice dell'algoritmo di Bellman-Ford. Si fornisca una stima asintotica del tempo di esecuzione dell'algoritmo nel caso pessimo giustificando con chiarezza la risposta.
  
- c) Si disegni la tabella  $M$  computata dall'algoritmo per minimum coin change problem quando la banconota da cambiare ha valore 6 e si hanno a disposizione monete con i seguenti tre valori: 1, 2, 3. Alla fine, fornire la soluzione ottima.

**4. Analisi degli algoritmi e notazione asintotica [20 minuti]**

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

1.  $n^2 \log^2 n + 10n^3 = O(n^2 \log^3 n)$
2.  $n^{1/3} = \Omega(n^{1/3} + \log n)$
3.  $n^{1/3} = \Omega(n^{1/3} + n^{1/4} \log n)$
4.  $n^3 - 10n^2 + 8 = \Omega(n^3 + 1000n^2 + n + 11)$
5.  $\log(\log^3 n) = O((\log n)(\log n))$

b) Dimostrare che la seguente affermazione e' vera giustificando la risposta. Occorre fornire le costanti  $c$  ed  $n_0$ .

1.  $2n^3 + 5n = O(n^3)$

c) Dimostrare che la seguente affermazione e' falsa giustificando **matematicamente** la risposta.

2.  $n^3 = \Omega(n^3 \log n)$

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<=2^n; j=j*2) {  
    print(j);  
  }  
}
```

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

- a) Si scriva lo pseudocodice di un algoritmo ricorsivo che prende in input un array (**ed eventualmente altri input**) e computa la lunghezza della sottosequenza che contiene il massimo numero di occorrenze consecutive di uno stesso elemento (l'elemento **NON e' passato in input**). È sufficiente che l'algoritmo abbia tempo  $O(n \log n)$ . **Evitate di invocare algoritmi ausiliari (punteggio inferiore se cio' viene fatto).**  
**Bonus** se l'algoritmo è lineare.
  
- b) Si fornisca la relazione di ricorrenza che esprime il limite superiore al tempo di esecuzione dell'algoritmo al punto a) . Si giustifichi la relazione fornita.
  
- c) A partire dalla relazione di ricorrenza da voi fornita al punto b) per il caso pessimo, si fornisca una funzione  $h(n)$  tale che  $T(n)=O(h(n))$ . **Giustificare la risposta** usando il metodo iterativo o quello della sostituzione (induzione).

**Domande aggiuntive per l'esame da 6 CFU.**

1.

c) Si fornisca l'algoritmo **ricorsivo** che prende in input un DAG  $G$  e restituisce l'ordinamento topologico di  $G$ . Fornire una stima del tempo di esecuzione dell'algoritmo nel caso pessimo giustificando la risposta.

2.

d) Si fornisca un'istanza del problema della minimizzazione dei ritardi con  $n=5$  per cui il valore della soluzione ottima è 7 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 7 e qual è il ritardo di tutte le attività.

3.

d) Fornire la tabella costruita dall'algoritmo che computa il valore della soluzione ottima per **subset sums** quando l'istanza input è  $w_1=1, w_2=2, w_3=7, w_4=5, w_5=4, W=6$ .