Algoritmi e Laboratorio
a.a. 2004/2005

Prova scritta del 7 gennaio 2005

COGNOME ……………………… NOME ………..………………

Matr. n. …………………..

1. **(PUNTI 2 + 2)** Completare i seguenti due algoritmi con gli <u>invarianti di ciclo</u> che permettono di dimostrarne la correttezza.

<u>ALGORITMO 1</u>
{A.I.: $n \geq 0$ & A[0..n] contiene $a_0 \ldots a_n$ }
y ← 1
result ← A[0]
i ← 1

{I.C. : ………………………………………………………………………………………}
while i ≤ n **do**
  y ← y • x
  result ← result +A[i] • y
  i ← i + 1

return result
{A.F.: result = A[0] + A[1] • x + ... + A[n] • x^n }

<u>ALGORITMO 2</u>
{A.I.: $n \geq 0$ & A[0..n] contiene $a_0 \ldots a_n$ }
result ← A[n]
i ← n – 1

{I.C.: ………………………………………………………………………………………}
while i ≥ 0 **do**
  result ← result • x + A[i]
  i ← i – 1

return result
{A.F.: result = A[0] + A[1] • x + ... + A[n] • x^n }

2. **(PUNTI 1+2+1)**

a) Fornire un lower bound al numero di confronti richiesti per ordinare n oggetti nel modello basato su confronti.
b) Fornire una traccia della dimostrazione.
c) Indicare il nome di due algoritmi ottimi.

3. **(PUNTI 2+1)**
Risolvere la seguente equazione di ricorrenza:

$T(1) = 2$	per n = 0 o n=1
$T(n) = 3T(\lfloor n/2 \rfloor) + T(\lfloor n/2 \rfloor) + \lfloor n/3 \rfloor$	per n >1

- mediante il metodo iterativo
- mediante applicazione del teorema principale

4. (PUNTI 1+4)

a) Scrivere lo "schema generale" dell'algoritmo Divide-et-impera.

b) Applicarlo per trovare un algoritmo di soluzione per il seguente problema:
Date due sequenze entrambe di n elementi ($n \geq 1$), memorizzate in due array A e B, restituire il valore: (A[1] * B[1]) + (A[2] * B[2]) + … + (A[n] * B[n])

5. (PUNTI 1 + 2 + 1)

a) Fornire l'algoritmo *Greedy-Activity-Selector* .

b) Applicare l'algoritmo *Greedy-Activity-Selector* al seguente insieme di attività, di cui sono forniti gli intervalli di attività [s, f).

	s	**f**
A	**5**	**9**
B	**4**	**6**
C	**6**	**8**
D	**7**	**8**
E	**7**	**10**
F	**3**	**9**
G	**5**	**7**
H	**4**	**9**

c) L'insieme massimale di attività mutuamente compatibili trovato è l'unico per l'insieme di attività dato? (MOTIVARE LA RISPOSTA.)

6. (PUNTI 1+2) Sia dato il grafo non orientato rappresentato dalla seguente lista di adiacenza:

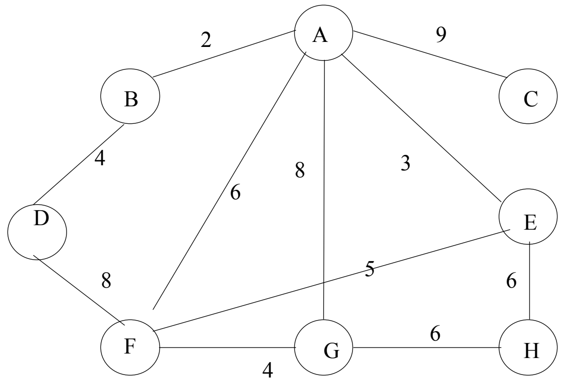
A	B C E G
B	A D
C	A
D	B F
E	A F H
F	D E G
G	A F H
H	E G

a) se ne effettui la visita in profondità, considerando A come vertice sorgente, e si indichino per ogni vertice i tempi di inizio e fine visita.

b) si disegni il grafo, etichettando ogni suo arco come T (dell'albero), B (all'indietro), F (in avanti), C (di attraversamento), avendo sempre A come vertice sorgente.

7. (PUNTI 5 + 1)

a) Si determini, applicando l'algoritmo di Prim a partire dal vertice A, un minimum spanning tree del seguente grafo non orientato, connesso, e pesato.



Per svolgere correttamente l'esercizio occorre:

- compilare le seguenti tabelle (la riga 0 è già compilata), e
- disegnare (un disegno per ogni riga delle tabelle) l'albero mantenuto dall'algoritmo (contenente SIA gli archi che fanno parte della soluzione CHE gli archi candidati) al termine di ogni iterazione (cioe' DOPO che sono state aggiornate le appetibilita' dei vertici in coda).

La prima tabella indica, per ogni iterazione del ciclo esterno, per ogni vertice v che non appartiene alla soluzione (l'albero definitivo), il peso di un arco di peso minimo che collega tale vertice ad un vertice della soluzione (∞ **se tale arco non e' ancora stato trovato)**. Il valore di k non deve essere più riportato quando il vertice e' ormai parte della soluzione (si metta il simbolo -).

La seconda tabella indica, per ogni iterazione del ciclo esterno, l'insieme dei vertici v che sono entrati a far parte della soluzione (per i quali k[v] è il peso dell'arco incluso).

La riga 0 corrisponde al temine dell'inizializzazione (prima di entrare nel ciclo).

Quando nella coda con priorità ci sono vertici con lo stesso valore minimo di k si scelga quello che viene prima secondo l'ordine alfabetico.

k	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
0	**0**	∞	∞	∞	∞	∞	∞	∞
1								
2								
3								
4								
5								
6								
7								
8								

	Insieme dei vertici t inclusi nella soluzione
0	{ }
1	
2	
3	
4	
5	
6	
7	
8	

c) L'albero ottenuto e' l'unico albero di copertura minimo per il grafo dato? (MOTIVARE LA RISPOSTA.)

8. **(PUNTI 4)** Sia G un grafo non orientato e connesso e CODICE un array (indicizzato sui vertici di G) che contiene, per ogni vertice v, un numero naturale CODICE[v]. Scrivere un algoritmo ricorsivo di visita in profondità a partire da un vertice sorgente s dato, che restituisca:

- TRUE se, per ogni vertice v del grafo, il codice di v è uguale alla somma dei codici degli antenati di v nell'albero di copertura ottenuto con la visita.
- FALSE, altrimenti.