

**Laurea Specialistica in “Informatica”**  
**Corso di “Algoritmi e Strutture Dati”**  
**Appello del 1° Giugno 2004**

1. *Tempo disponibile 180 minuti (è ammesso ritirarsi entro 90 minuti)*
2. *Sono ammessi al più 3 scritti consegnati per A.A.*
3. *Non è possibile consultare appunti, libri o persone, né uscire dall'aula*
4. *Nella valutazione dello scritto ogni esercizio conta 6 punti (e quindi si raggiunge 18 con 3 esercizi risolti correttamente e 30 con 5 esercizi risolti correttamente)*
5. *Le soluzioni degli esercizi devono:*
  - a. *Spiegare a parole l'algoritmo usato (anche con eventuali disegni)*
  - b. *commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)*
  - c. *giustificare la correttezza e tutti i passaggi matematici*
  - d. *dimostrare la complessità (con equazioni di ricorrenza se necessario)*

1. Si valuti l'ordine di grandezza della complessità  $T(n)$  della seguente funzione Pascal:

```
function MISTERO(n: integer): integer;  
  var j,k: integer;  
  begin  
    for j := n - 11 to n do k := n * j;  
    j := 2;  
    if n > 33 then  
      MISTERO := ( 3 * MISTERO(n div j) + MISTERO(31) ) div 2  
    else  
      MISTERO := 17;  
  end;
```

2. Data una sequenza  $a_1, \dots, a_n$  di  $n$  interi, si vuole calcolarne la sommatoria  $a_1 + \dots + a_n$ . Si scriva una procedura (o funzione) Pascal efficiente che utilizzi la tecnica *divide-et-impera con partizione bilanciata dei dati*, assumendo che la sequenza sia realizzata con un vettore. Se ne analizzi la complessità impostando e risolvendo le opportune relazioni di ricorrenza.

3. Data una lista  $L$  di interi, si vuole modificarla cancellando tutti gli elementi adiacenti che hanno valori consecutivi, mantenendo lo stesso ordine che gli elementi avevano inizialmente (p.e. se l'ingresso è  $L = \underline{6}, \underline{5}, \underline{6}, 4, 1, \underline{1}, \underline{2}, \underline{8}, \underline{7}, 4$  allora il risultato è  $L = 4, 1, 4$ ). Si scriva una procedura Pascal efficiente *utilizzando gli operatori* per le liste visti a lezione.

4. Si scriva la procedura Pascal *Depth-First Search (DFS)* vista a lezione. Si esegua la procedura DFS sul grafo *non orientato*  $G = (N, A)$ ,  $N = \{1, 2, 3, 4, 5\}$ ,  $A = \{[1,2], [1,3], [1,5], [2,5], [3,4], [3,5]\}$  a partire dal nodo 2, assumendo che i vettori di adiacenza siano ordinati in modo *crescente* e mostrando il contenuto dei vettori di adiacenza.

5. Si scriva la procedura HEAPSORT vista a lezione e la si esegua (a mano) per ordinare alfabeticamente gli 11 elementi: O, R, D, I, N, A, M, E, N, T, O. Si illustri con disegni, passo dopo passo, il contenuto dello heap durante l'esecuzione.

6. Si indichi il contenuto di una tabella hash di dimensione 15, inizialmente vuota, dopo l'inserzione, nell'ordine, delle chiavi: T, R, A, P, A, G, L, I, E, U, R, O, P, E, I. Si usi la funzione hash  $H(k) = k \bmod 15$  per la  $k$ -esima lettera dell'alfabeto italiano e il metodo di scansione lineare a passo unitario. Si indichi successivamente il contenuto della tabella dopo avervi cancellato, nell'ordine: G, I, L, A, R, D, I, N, O, e poi inserito, nell'ordine: B, A, G, G, I, O.