

**Laurea in “Scienze di Internet”**  
**Corso di “Algoritmi e Strutture Dati”**  
**21 Settembre 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 PIPPO(n: integer): integer;  
  var j, k: integer;  
  begin  
    for j := 2 to 4 do k := j;  
    if n > 7 then  
      PIPPO := n * PIPPO(n div k) + PIPPO(k div k) div k  
    else  
      PIPPO := 7;  
  end;
```

2. Dato un albero binario  $T$  contenente elementi interi, lo si vuole modificare cancellando ogni foglia che contiene un elemento uguale a quello del padre. Si scriva una procedura Pascal di complessità ottima assumendo che l'albero sia *realizzato con puntatori*.

3. Data una lista  $L$  di interi, si vuole modificarla cancellando tutti gli elementi adiacenti che hanno valori identici, mantenendo lo stesso ordine che gli elementi avevano inizialmente (p.e. se l'ingresso è  $L = \underline{6}, \underline{6}, \underline{6}, 2, 3, \underline{1}, \underline{1}, \underline{7}, \underline{7}, 4$  allora il risultato è  $L = 2, 3, 4$ ). 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 3, assumendo che i vettori di adiacenza siano ordinati in modo *crescente* e mostrando il contenuto dei vettori di adiacenza.

5. Nel problema delle SOMME PREFISSE, data una sequenza  $a_1, \dots, a_n$  di interi, si vuole calcolare la sequenza  $b_1, \dots, b_n$  dove  $b_i$  è uguale alla somma  $a_1 + \dots + a_i$ . Si scriva una procedura Pascal che utilizzi la tecnica *divide-et-impera con partizione bilanciata dei dati*, assumendo che le sequenze siano realizzate con *vettori*, e se ne analizzi la complessità.

6. Si consideri il problema dell'INSIEME DOMINANTE: "Dati un grafo non orientato  $G=(N,A)$  ed un intero  $k$ , esiste un sottoinsieme  $S$  di al più  $k$  nodi tale che ciascun nodo in  $N - S$  sia adiacente ad almeno un nodo in  $S$ ?" Si scriva un algoritmo *non deterministico* di complessità polinomiale.