

Algoritmi e Strutture Dati

19 Febbraio 2003

- *Tempo disponibile 180 minuti (è ammesso ritirarsi entro 90 minuti)*
- *Sono ammessi al più 3 scritti consegnati per A.A.*
- *Non è possibile consultare appunti, libri o persone, né uscire dall'aula*
- *Le soluzioni degli esercizi devono:*
 - *Spiegare a parole l'algoritmo usato (anche con eventuali disegni)*
 - *commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)*
 - *giustificare la correttezza e tutti i passaggi matematici*
 - *dimostrare la complessità (con equazioni di ricorrenza se necessario)*
- *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)*

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

```
function MISTERO(n: integer): integer;  
  var i, j: integer;  
  begin  
    for i := 0 to 1 do j := 2;  
    if n > 10 then  
      MISTERO := 5*MISTERO(n div 2) + MISTERO(n div j) + j*n div 2  
    else if n > 2 then  
      MISTERO := 5*MISTERO(n div 2) + MISTERO(n - j) + j*n div 2  
    else  
      MISTERO := 1  
  end;
```

2. Si consideri una nuova operazione delle liste che, data una lista L di interi, la modifica duplicando tutti gli elementi divisibili per 3 mantenendo lo stesso ordine che gli elementi avevano in L (p.e. se l'ingresso è $L = 6, 1, 3, 4$, allora il risultato è $L = 6, 6, 1, 3, 3, 4$). Si scriva una procedura Pascal di complessità ottima *utilizzando gli operatori* per le liste visti a lezione.

3. Si consideri una nuova operazione degli alberi binari che, dato un albero binario T contenente elementi interi, lo modifica cancellando ogni foglia che contiene un elemento pari. Si scriva una procedura Pascal di complessità ottima assumendo che l'albero sia *realizzato con puntatori*.

4. Si riscriva la procedura Pascal *Depth-First Search (DFS)* vista a lezione e la si esegua sul grafo non orientato $G=(N,A)$, $N=\{1,2,3,4,5\}$, $A=\{[1,2],[1,3],[1,5],[3,5],[3,4],[4,1],[4,2]\}$ a partire dal nodo 5. Si disegni il grafo e si mostrino la memorizzazione di G con vettori di adiacenza e l'ordine di visita dei nodi e degli archi (assumendo che i vettori di adiacenza siano ordinati in modo *crescente*).

5. Dati un vettore ordinato di n interi distinti e due interi s e d , con $s < d$, si vuole decidere se esistono o no elementi del vettore che appartengono all'intervallo aperto (s,d) . Si scriva una procedura (o funzione) Pascal *divide et impera* modificando la *ricerca binaria* vista a lezione.

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