

# Laurea in “Informatica”

## Corso di “Algoritmi e Strutture Dati”

### 19 Settembre 2006

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati tra Giugno 2006 e Febbraio 2007.
3. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
4. Per raggiungere la sufficienza occorrono almeno 3 esercizi risolti senza alcun errore.
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 la complessità  $T(n)$  della seguente funzione Pascal:

```
function PIPPO(n: integer): integer;  
  var i, j: integer;  
  begin  
    for j := n - 2 to n + 4 do i := n;  
    if n < 17  
      then PIPPO := n  
      else if n > 113  
        then PIPPO := 3*PIPPO(n div i) + 3*PIPPO(n div 2) + n*i  
        else PIPPO := 3*PIPPO(n - 2) + 3*i;  
  end;
```

2. Sia dato un albero binario  $A$  implementato con puntatori, in cui ogni nodo contiene un elemento intero. Scrivere una funzione Pascal di *complessità ottima* che modifichi  $A$  cancellandone tutte le foglie che contengono un elemento pari.

3. In una lista  $L$  di interi, un *massimo locale* è un elemento maggiore sia del suo predecessore sia del suo successore. Si scriva una procedura Pascal di *complessità ottima*, utilizzando gli operatori visti a lezione, per modificare  $L$  in modo che contenga, nello stesso ordine, tutti e soli i massimi locali (p.e. se in input  $L = 3, 7, 8, 1, 4$ , allora in output  $L = 8, 4$ ).

4. Sia dato un vettore  $V$  contenente  $n$  interi positivi distinti. Si consideri l'operatore  $INV(x)$  che calcola in tempo costante l'intero ottenuto invertendo l'ordine delle cifre di  $x$  (p.e. se  $x = 537$ ,  $INV(x) = 735$ ). Si scriva una procedura Pascal efficiente che verifichi se esistono in  $V$  due elementi distinti  $x$  ed  $y$  tali che  $y = INV(x)$ .

5. Dato l'insieme  $I = \{11, 3, 8, 2, 1, 9, 6, 10, 12, 5, 4, 7, 13\}$ , indicare la sua rappresentazione mediante: (1) albero binario di ricerca con altezza minima; (2) albero binario di ricerca con altezza massima; (3) albero 2-3; (4) heap; (5) mset.

6. Dati un insieme  $A$ , di  $n$  interi distinti, ed un intero  $k$ , si vuole decidere se esiste un sottoinsieme  $S$  di  $A$  tale che il prodotto degli elementi in  $S$  sia uguale a  $k$ . Scrivere (in pseudocodice) un algoritmo non deterministico che richieda tempo polinomiale.