

Algoritmi e Strutture Dati

13 Gennaio 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 BOH(n: integer): integer;  
  var i, j, k, m: integer;  
  begin  
    m := n;  
    while m > 1 do begin  
      for j := 1 to n do k := n * m * j;  
      m := m - 1  
    end;  
    if n > 33 then  
      BOH := n * BOH(n div 2) + n  
    else  
      BOH := 1;  
  end;
```

2. Sia dato un vettore *ordinato* $V[1..n]$ contenente *tutti* gli $n-1$ elementi interi distinti appartenenti all'intervallo $1..n-1$. Si scriva una procedura (o funzione) Pascal, basata sulla ricerca binaria, per individuare in tempo $O(\log n)$ l'unico intero dell'intervallo $1..n-1$ che compare due volte in V .

3. Data una lista L di interi, si vuole modificarla cancellando tutti gli elementi pari e duplicando tutti gli elementi dispari, mantenendo lo stesso ordine che gli elementi avevano inizialmente (p.e. se l'ingresso è $L = 6, 1, 4, 3, 5$ allora il risultato è $L = 1, 1, 3, 3, 5, 5$). Si scriva una procedura Pascal di complessità ottima *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 sul grafo orientato $G = (N, A)$, $N = \{1, 2, 3, 4, 5\}$, $A = \{(1,2), (1,3), (1,5), (3,5), (3,4), (4,1), (4,2), (5,2)\}$ a partire dal nodo 4, assumendo che i vettori di adiacenza siano ordinati in modo *crescente* e mostrando il contenuto dei vettori di adiacenza.

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

6. Dato un insieme A di interi, si vuole decidere se esiste una partizione di A in due sottoinsiemi B e C tali che la somma degli elementi di B sia uguale al prodotto degli elementi di C . Si scriva una procedura Pascal *non deterministica* di complessità polinomiale.