

ALGORITMI E STRUTTURE DATI – SCIENZE DI INTERNET

14 Gennaio 2003

Esercizio 1

$$T(n) = \begin{cases} c_1 & n \leq 44 \\ T(n-2) + nc_2 & n > 44 \end{cases} \quad c_1, c_2 \text{ costanti}$$

$$a=1 \quad \beta=1 \quad T(n) \text{ è } O(n^{\beta+1}) = O(n^2)$$

Esercizio 2

Usare gli operatori visti a lezione!

```
Procedure paridispari(var L, M:liste);
var p, q:posizione;
begin
  p:=primolista(L);
  crealista(M);
  q:=primolista(M);
  while not finelista(p,L) do
    if (leggilista(p,L) mod 2 = 0) then
      canclista(p,L);
    else begin
      inslista(leggilista(p,L),q,M);
      p:=succlista(p,L);
      q:=succlista(q,M)
    end
  end
end
```

Complessità: $O(n)$

Esercizio 3

Usare i puntatori!

```
Procedure cancellafoglie(var T:albero, n:nodo);
begin
  if (u^.destro = nil) and (u^.sinistro = nil) and (u^.valore mod 2 = 0) then begin
    if u^.genitore = nil then T:=nil
    else if u^.genitore^.sinistro = u then
      u^.genitore^.sinistro := nil
    else u^.genitore^.destro := nil;
    dispose(u)
  end
  else begin
    if u^.sinistro <> nil then cancellafoglie(T, u^.sinistro);
    if u^.destro <> nil then cancellafoglie(T, u^.destro)
  end
end
```

Complessità: $O(n)$

Esercizio 4

vettori di adiacenza (in grigio)

NODI	
1	1
2	4
3	4
4	6
5	8
6	9

ARCHI		
1	2	
2	3	A(1)
3	5	
4	4	
5	5	A(3)
6	1	
7	2	A(4)
8	2	A(5)

Ordine visita: **4** (4,1)(4,2) **1** (1,2)(1,3)(1,5) **2 3** (3,4)(3,5) **5** (5,2)

Esercizio 5

```
Procedure sommeprefisse(var A,B: vettore; i,j : integer);
var m, k: interger;
begin
  if i=j then    B[i] := A[i];
  else begin
    m := (i+j) div 2;
    sommeprefisse(A, i, m);
    sommeprefisse(A, m+1, j);
    for k:=m+1 to j do
      B[k] := B[k] + B[m]
    end
  end
end
```

Complessità:

$$T(n) = \begin{cases} c_1 & n = 1 \\ 2T(n/2) + nc_2 & n > 1 \end{cases} \quad c_1, c_2 \text{ costanti}$$

Ipotesi $n=2^h$

$$\alpha = (\log a)/(\log b) = (\log 2)/(\log 2) = 1 = \beta$$

$T(n)$ è $O(n \log n)$

NOTA: non era richiesta complessità ottima

Esercizio 6

```
Procedure partizione(var A: vettore; k : integer);
var   B, C : packet array [1..n] of boolean;
      i, sommaB, sommaC, prodD : integer;
begin
  for i:=1 to n do begin
    B[i] := choice {true, false};
    if not B[i] then C[i] := choice{true, false}
  end;
  sommaB:= 0;
  sommaC:= 0;
  prodD := 1;
  for i:=1 to n do
    if B[i] then sommaB := sommaB + A[i];
    else if C[i] then
      sommaC := sommaC + A[i]
    else
      prodD := prodD * A[i];
    if (sommaB = k) and (sommaC = prodD) then
      success else failure
  end
end
```

Complessità: $O(n)$