

ALGORITMI E STRUTTURE DATI – SCIENZE DI INTERNET

13 Gennaio 2004

Esercizio 1

$$T(n) = \begin{cases} c_1 n^2 & n \leq 33 \\ T(n/2) + c_2 n^2 & n > 33 \end{cases} \quad c_1, c_2 \text{ costanti}$$

Per il teorema delle ricorrenze lineari con partizione bilanciata:

$$\begin{aligned} a=1 \quad b=2 \quad \alpha &= (\log a)/(\log b) = 0 & \beta &= 2 \\ \beta > \alpha & \text{ quindi } T(n) \text{ è } O(n^\beta) = O(n^2) \end{aligned}$$

Esercizio 2

```
Function TROVADOPPIO(primo, ultimo: integer; var V:vettore) : integer;
var   medio : integer;
begin
  if (primo < ultimo) then begin
    medio := (primo + ultimo) div 2;

    if medio > V[medio] then
      TROVADOPPIO := TROVADOPPIO(primo, medio-1, V);
    else
      TROVADOPPIO := TROVADOPPIO(medio+1, ultimo, V)
    end
  else
    TROVADOPPIO := V[primo]
  end;
end;
```

Funzione basata sulla ricerca binaria, complessità: $O(\log n)$

Esercizio 3

Utilizzare gli operatori per le liste visti a lezione!

```
Procedure CANCELLAEDUPLICA(var L:lista);
var   p : posizione;
      temp : integer;
begin
  p := PRIMOLISTA(L);

  while not FINELISTA(p, L) do begin
    temp := LEGGILISTA(p, L);
```

```

    if (temp mod 2 = 1) then begin
        p := SUCCLISTA(p, L);
        INSLISTA(temp, p, L);
        p := SUCCLISTA(p, L)
    end else
        CANCLISTA(p, L)
    end
end;

```

Complessità ottima: $O(n)$

Esercizio 4

vettori di adiacenza (in grigio)

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

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

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

Esercizio 5

Procedure CANCELLANODI(u:nodo; Var T:binalbero);

begin

 if (u^.sinistro = NIL) and (u^.destro = NIL) then begin

 if (u^.genitore <> NIL) then begin

 if (u^.genitore^.valore = u^.valore) then begin

 if (u^.genitore^.sinistro = u) then

 u^.genitore^.sinistro := NIL

 else

 u^.genitore^.destro := NIL;

 dispose(u)

 end

 end

 end else begin

 if (u^.sinistro <> NIL) then CANCELLANODI(u^.sinistro, T);

 if (u^.destro <> NIL) then CANCELLANODI(u^.destro, T)

 end

end;

Complessità: $O(n)$

Esercizio 6

```
Procedure partizione(var A: vettore);
var   B : packet array [1..n] of boolean;
      i, sommaB, prodC : integer;
begin
  for i:=1 to n do B[i] := choice {true, false};

  sommaB:= 0;
  prodC := 1;

  for i:=1 to n do
    if B[i] then   sommaB := sommaB + A[i];
    else           prodC := prodC * A[i];

    if (sommaB = prodC) then success
    else failure
  end
end
```