

## Algoritmi e Strutture Dati

### Esercizio 1.[11 punti]

Calcolare in ordine di grandezza il costo computazionale della funzione BIG. Dire quanto vale esattamente  $BIG(n)$  in funzione di  $n$ . Giustificare sempre tutte le risposte.

BIG(n)

```
1  if  $n \leq 1$ 
2      return 1
3  else
4      return 2 BIG( $n - 1$ )
```

### Traccia Soluzione esercizio 1

$$T(n) = T(n - 1) + k =$$

$$T(n - 2) + k + k =$$

$$T(n - 3) + k + k + k =$$

$\vdots$

$$\text{E quindi } T(n) = \Theta(n)$$

BIG( $n$ ) restituisce sempre  $2^{n-1}$ .

Dimostrazione:

- Per  $n = 1$  è banalmente vero

- Se è vero (ipotesi induttiva) che  $BIG(n - 1) = 2^{n-2}$  allora  $BIG(n) = 2 \text{ BIG}(n - 1) = 2 \cdot 2^{n-2} = 2^{n-1}$

### Esercizio 2.[11 punti]

Sia  $T$  un qualunque albero ternario (ogni nodo ha al massimo tre figli chiamati "primo", "secondo" e "terzo") i cui nodi contengono, nel campo "valore", un numero intero maggiore o uguale a 0. Scrivere una procedura ricorsiva MODIFICA che prenda in input  $T$  e lo modifichi seguendo le seguenti regole.

Per ogni nodo  $p$  dell'albero:

- se  $p$  contiene il valore 0, i figli di  $p$  non vengono modificati
- se  $p$  contiene il valore 1, il sottoalbero radicato nel primo figlio di  $p$  e il sottoalbero radicato nel terzo figlio di  $p$  vengono scambiati tra loro

- se  $p$  contiene il valore 2, il sottoalbero radicato nel secondo figlio di  $p$  viene eliminato
- se  $p$  contiene un valore diverso da 0,1 e 2 tutti i discendenti di  $p$  vengono eliminati.

Discutere la correttezza di MODIFICA e il suo il costo computazionale.

## Traccia Soluzione esercizio 2

MODIFICA( $p$ )

```

1  if  $p \neq \text{NIL}$ 
2      if  $p.\text{valore} > 2$ 
3           $p.\text{primo} = \text{NIL}$ 
4           $p.\text{secondo} = \text{NIL}$ 
5           $p.\text{terzo} = \text{NIL}$ 
6      if  $p.\text{valore} == 1$ 
7           $\text{temp} = p.\text{primo}$ 
8           $p.\text{primo} = p.\text{terzo}$ 
9           $p.\text{terzo} = \text{temp}$ 
10     if  $p.\text{valore} == 2$ 
11          $p.\text{secondo} = \text{NIL}$ 
12     MODIFICA( $p.\text{primo}$ )
13     MODIFICA( $p.\text{secondo}$ )
14     MODIFICA( $p.\text{terzo}$ )

```

Costo  $\Theta(n)$ .

## Esercizio 3.[11 punti]

Sia  $S = \{s_1, \dots, s_n\}$  un insieme di segmenti. Ogni segmento  $s_i$  è una coppia di numeri reali  $(l_i, r_i)$ .  $l_i$  e  $r_i$  rappresentano rispettivamente gli estremi sinistro e destro del segmento  $i$ -esimo.

Fornire un algoritmo per selezionare da  $S$  un sottoinsieme  $T \subseteq S$  di segmenti di cardinalità massima che soddisfino la seguente proprietà :

per ogni coppia di segmenti  $(l_i, r_i)$  e  $(l_j, r_j)$  di  $T$ , deve valere

$$(r_i < l_j) \text{ oppure } (r_j < l_i)$$

ovvero i due segmenti non si sovrappongono (nemmeno parzialmente).

Discuterne la correttezza e il costo computazionale.

## Traccia Soluzione esercizio 3

Il problema è banalmente riconducibile al problema di "selezione di attività " visto a lezione