Soluzioni

Esercizio 1

Somma (a, sx, dx)

```
if (sx > dx) return 0;
if (sx == dx) return a[sx];
cx = (sx+dx)/2
return Somma(a, sx, cx) + Somma(a, cx+1, dx)
```

1) $$T(n) = \begin{cases} \Theta(1) & n \leq 1 \\ 2T\left(\frac{n}{2}\right) + \Theta(1) & n > 1 \end{cases}$$

$a = 2$

$b = 2$ $\qquad n^{\log_b a} = n^1$

$f(n) = \Theta(1)$

$f(n) = O\left(n^{\log_b a - \varepsilon}\right) \qquad 0 < \varepsilon < 1$

Teorema dell'esperto, I° caso

$$T(n) = \Theta\left(n^{\log_b a}\right) = \Theta(n)$$

2)

$$T(n) = \begin{cases} \Theta(1) & n \leq 1 \\ 2\,T\left(\frac{n}{2}\right) + O(\log n) & n > 1 \end{cases}$$

$a = 2$

$b = 2 \qquad n^{\log_b a} = n^1$

$f(n) = O(\log n)$

$f(n) = O\left(n^{\log_b a - \varepsilon}\right) \qquad 0 < \varepsilon < 1$

Teorema dell'esperto, I° caso:

$$T(n) = \Theta\left(n^{\log_b a}\right) = \Theta(n)$$

ESERCIZIO 2

```
Mink ( H, i, k)
if ( i ≤ H. heapsize) {
    if ( H[i] ≤ k) {
        Print H [i];
        Mink ( H, Left (i), k);
        Mink ( H, Right (i), k);
    }
}
```

L'algoritmo visita lo heap, ma si ferma non appena la chiave della radice del sottoalbero è $> k$.

Dunque $T(n) = \Theta(m)$

dove n è la dimensione dello heap, e m sono gli elementi di chiave $\leq k$.

ESERCIZIO 3

Sia T un albero binario di n nodi.

Larghezza (T)

```
if (T.root == NIL) return 0;
larghezza = 1
livello = 0   // il livello della radice è 0
ctr = 1 // sul livello 0 c'è un nodo
u = T.root;
u.d = 0;
Q = nuova coda
Enqueue (Q, u);
while (Q non è vuota) {
    v = Dequeue (Q);
    if (v.d ≠ livello) {  // siamo scesi di un livello
        livello = v.d;
        ctr = 0;
    }
```

```
ctr ++;
if (ctr > larghezza) larghezza = ctr;
if ( v.left ≠ NIL) {
    v.left.d = v.d + 1;
    Enqueue (Q, v.left);
}
if (v.right ≠ NIL) {
    v.right.d = v.d + 1;
    Enqueue (Q, v.right);
}
}
return larghezza;
```

ANALISI

L'algoritmo esegue una visita per livelli dell'albero.
La complessità in tempo è

$$T(n) = \Theta(n).$$

La complessità in spazio è

$$S(n) = \Theta(n),$$

ed è dovuta all'uso della coda Q, e al campo "d" attribuito a ciascun nodo dell'albero.

ESERCIZIO 4

$$T(i) = \begin{cases} \text{Costo di Stampa}(B) & i = n \\ 2T(i+1) + \underbrace{c}_{\text{costante}} & 0 \leq i < n \end{cases}$$

$$T(0) = 2T(1) + c = 2\left[2T(2) + c\right] + c =$$

$$= 2^2 T(2) + 3c = 8\,T(3) + 7c =$$

$$= \dots = 2^i\, T(i) + (2^i - 1) \cdot c$$

$$\underset{i=n}{=} 2^n\, T(n) + (2^n - 1)\, c =$$

$$= 2^n \cdot \text{Costo di Stampa}(B) + (2^n - 1) \cdot c$$

$$= \Theta\left(2^n \cdot \text{Costo Stampa}(B)\right)$$