

# Seconda Prova di Verifica Intermedia

3 giugno 2019

## ESERCIZIO 1

Si visita il grafico per verificare se è connesso e se tutti i vertici hanno grado pari.

### Euleriano (G)

for all  $v \in V$

$v$ .color = B;

scegli  $s \in V$  come sorgente;

$s$ .color = G;

$Q$  = nuova coda ();

Enqueue ( $Q, s$ );

while ( $Q \neq \emptyset$ ) {

$u$  = Dequeue ( $Q$ );

grado = 0;

for all  $v \in \text{Adj}(u)$  {

    grado ++;

    if ( $v$ .color == B) {

$v$ .color = G;

        Enqueue ( $Q, v$ );

    }   
 }   
 if (grado % 2 == 1) return FALSE;

// il vertice  $u$  non ha  
// grado pari

}

... continue ...



```

for all v in V {
  if (v-color == B) return FALSE;
  // il grafo non è connesso
}

```

```

return TRUE // il grafo è connesso e tutti i vertici
            // hanno grado pari

```

$$T(|V|, |E|) = O(|V| + |E|)$$

**Esercizio 2**

① AVL2array(T) // T: albero di n nodi

```

a = nuovo array di dimensione n
i = 0;
visita(T.root, a, &i)
return a;

```

visita(u, a, i) // i: pointer per riferimento

```

if (u != NIL) {
  visita(u.right, a, i)
  a[*i] = u.key; (*i)++;
  visita(u.left, a, i)
}

```

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

②

Si visita, ad esempio in ordine anticipato, l'albero AVL  $T_1$ , e si inseriscono le chiavi in  $T_2$ .

Merge ( $T_1, T_2$ )

visita ( $T_1, \text{root}, T_2$ )

return  $T_2.\text{root}$

visita ( $u, T_2$ )

if ( $u \neq \text{NIL}$ ) {

Insert AVL ( $T_2.\text{root}, u.\text{key}$ )

visita ( $u.\text{left}, T_2$ )

visita ( $u.\text{right}, T_2$ )

}

$$\begin{aligned}
 T(n, m) &= O(n \cdot \log(n+m)) \\
 &= O(n \log m) \\
 &\quad n < m
 \end{aligned}$$

3

Merge (T1, T2)

a = AVL2array(T1) // in senso crescente, lista simmetrica standard

b = AVL2array(T2) // in senso crescente

c = nuovo array di dimensione n+m

c = Merge(a, b) // fusione di a e b

return Array2AVL(c, 0, n+m-1);

Array2AVL(c, sx, dx)

if (sx > dx) return NULL;

cx =  $\frac{sx+dx}{2}$

u = nuovo nodo

u.key = c[cx];

u.left = Array2AVL(c, sx, cx-1)

u.right = Array2AVL(c, cx+1, dx);

return u;

Analisi

Sia k = n+m

Costo di Array2AVL:  $T(k) = 2T(\frac{k}{2}) + \Theta(1)$   
 $= \Theta(k)$

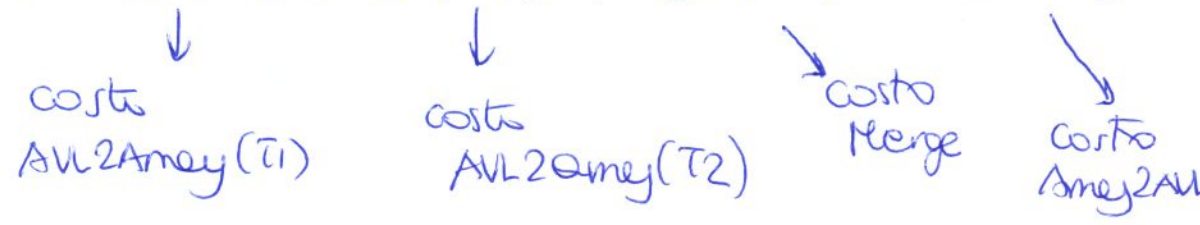
~~Costo Merge(T1, T2) =  $\Theta(n) + \Theta(m)$~~

Costo complessivo:

$$k = n + m$$

5

$$T(n, m) = \Theta(n) + \Theta(m) + \Theta(k) + \Theta(k)$$



$$= \Theta(n + m)$$

ESERCIZIO 3

|                  |   |   |   |   |   |   |   |
|------------------|---|---|---|---|---|---|---|
|                  |   |   | 1 | R | O | S | O |
|                  |   | 0 | 1 | 2 | 3 | 4 | 5 |
| A<br>R<br>S<br>O | 0 | 0 | 1 | 2 | 3 | 4 | 5 |
|                  | 1 | 1 | 1 | 2 | 3 | 4 | 5 |
|                  | 2 | 2 | 2 | 1 | 2 | 3 | 4 |
|                  | 3 | 3 | 3 | 2 | 2 | 2 | 3 |
|                  | 4 | 4 | 4 | 3 | 2 | 3 | 2 |

$$d_E(\text{ARSO}, \text{IROS O}) = 2$$

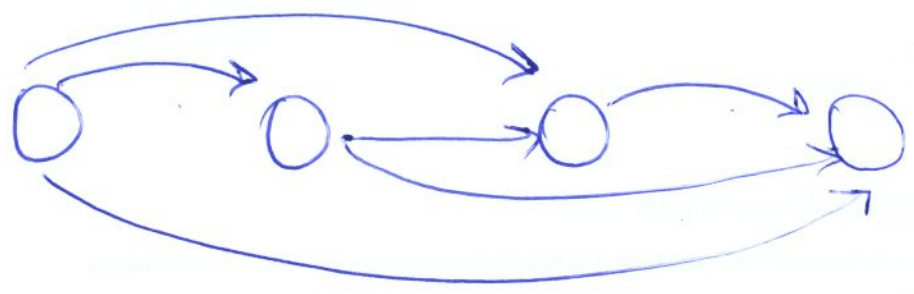
A R - S O  
I R O S O

+       +

ESERCIZIO 4

Si veda il libro di testo  
CLRS, cap 11

ESERCIZIO 5



6 archi