

## SOLUZIONI

Es. 1

1. Con l'Albero di decisione

$S(n) =$  le  $n-1$  posizioni possibili  $+ 1$  (nessun salto)  $= n$

$$2^t \geq S(n) \quad 2^t \geq n \quad t \geq \log_2 n$$

limite inferiore  $t = \Omega(\log n)$

2.

Salto  $(a, sx, dx)$ ;

if  $(sx == dx)$  return -1;

if  $(sx == dx - 1)$  if  $(|a[dx] - a[sx]| \geq 2)$   
return  $sx$ ;

else {

$cx = (sx + dx) / 2$ ;

if  $(|a[sx] - a[cx]| \geq cx - sx + 1)$   
return Salto  $(a, sx, cx)$ ;

else if  $(|a[dx] - a[cx]| \geq dx - cx + 1)$

return Salto  $(a, cx, dx)$

else return -1;

}

1.3  $\begin{cases} T(n) = c & \text{se } n \leq 2 \end{cases}$

$\begin{cases} T(n) = T(\frac{n}{2}) + c \end{cases}$

case 2 th. principle

$\Downarrow$

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

Es. 3

```
BFS (G, s, k);  
for ogni vertice  $u \in G - s$  {  
     $u.color = white$ ;  
     $u.d = \infty$ ;  
}  
 $s.d = 0$ ;  $stop = false$ ;  
 $Q = \emptyset$ ;  
while ( $Q \neq \emptyset$ ) and ( $!stop$ ) {  
     $u = dequeue(Q)$ ;  
    for ogni vertice  $v \in G.adj(u)$  {  
        if  $v.color == white$  {  
             $v.color = gray$ ;  
             $v.d = u.d + 1$ ;  
            if ( $v.d == k$ ) Print ( $v$ );  
            * if ( $v.d == k + 1$ )  $stop = true$ ;  
            Enqueue ( $Q, v$ );  
        }  
    }  
     $u.color = black$  }  
}
```

\* Quando s'incontra un nodo a distanza  $k+1$  la visita può terminare.

Es. 4.

1. Si basta individuare le posizioni dei 3 elementi con una scansione, cercando i minori del precedente e poi effettuare 3 passi di Insertion Sort o di Partition con l'elemento come ferro.
2. vedi libro
3. MSB ( $n \log n$ ) in qualsiasi. QS  $\Theta(n^2)$  pessimo.  $\Theta(n \log n)$  medio e ottimo