DOMANDA A

$$T(n) = \begin{cases} 2 & n=0 \\ T(n-1)+2n+1 & n>0 \end{cases}$$

$T(n)$	$2n+1$	← "parto da dimensione n e scrivo costo della parte non ricorsiva"
$T(n-1)$	$2(n-1)+1$	← " $n-1$
$T(n-2)$	$2(n-2)+1$	← " $n-2$

Suppongo sia quadratica $\Rightarrow T(n) = an^2+bn+c$

↓ dimostro per induzione

- $(n=0)$ $T(0) = a\cdot 0^2 + b\cdot 0 + c \rightarrow \boxed{c=2}$
- $(n>0)$ $T(n) = T(n-1)+2n+1 \overset{\text{ip. induttiva}}{=} a(n-1)^2+b(n-1)+c+2n+1$

$$= an^2-2an+a+bn-b+c+2n+1$$

$$= an^2+(b-2a+2)n+a+c+1-b$$

$$\overset{!}{=} an^2+bn+c$$

↓

$T(n) = n^2+2n+2$ ✓

DOMANDA C

LCS

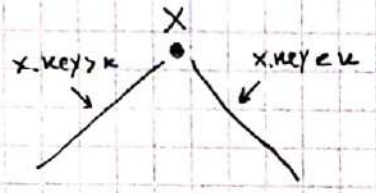
		O	R	O
	0	0	0	0
A	0	0	0	0
R	0	0	1	1
M	0	0	1	1
O	0	1	1	2

← inizializzo a 0

↑ inizializzo a 0

ESERCIZIO 1

Rep(T, k)



Non posso supporre che se ho x.key == k allora posso cercare solo a destra perché sarebbe un problema se avessimo che l'albero deve essere bilanciato!

```
RepRec (x, k) {
    if (x = nil)
        return 0
    else if x.key < k
        return RepRec (x.right, k)
    else if x.key > k
        return RepRec (x.left, k)
    else
        return 1 + RepRec (x.left, k) + RepRec (x.right, k)
}
```

```
Rep(T, k) {
    return RepRec(T.root, k)
}
```

sono nel caso in cui ho trovato il nodo con chiave uguale a k, devo ricordarmi quindi di averlo trovato (1+) e poi cerco a sinistra e a destra.

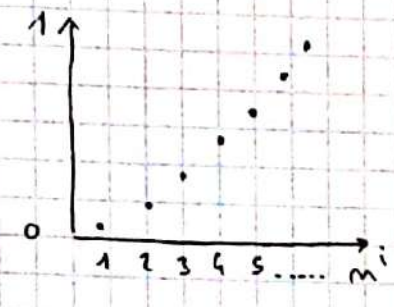
COMPLESSITÀ: $T(n) = O(n)$ nel caso pessimo (ovvero quando trovo la chiave k)

↓ Se volessi scriverlo per bene: $T(n) = c + T(n-l-1) + T(l)$ (radice)

↓ Soluzione lineare (equivalente ad una visita)

ESERCIZIO 2

GREEDY




```
n ← length (S)
first ← 1
last ← n
while ( first ≤ last ) {
    if ( first < last AND a.first + a.last ≤ 1 ) {
        P = P ∪ {{a.first, a.last}}
        first++
    }
    else
        P = P ∪ {{a.last}}
    last--
}
return P
```

L'algoritmo fa la scelta greedy

↳ $\{a_1, a_n\}$ se $n > 1$ e $a_1 + a_n \leq 1$

↳ $\{a_n\}$ altrimenti

DIMOSTRAZIONE (poco chiara, valeva max 2/3 punti)

OPT è un algoritmo che supponiamo non accoppi il più piccolo con il più grande suo accoppiamento

$a_1 \rightarrow \{a_1, a_2\}$

$a_n \rightarrow \{a_3, a_n\}$

⋮