Ejercicio 1

-3	-1	0	1	3	5	8	10
0	1	2	3	4	5	6	7

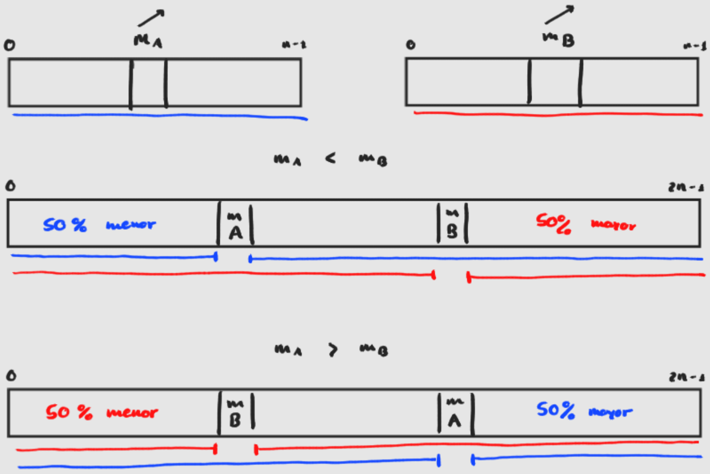
```
public int buscar(V, inf, sup) {
    med = (inf + sup)/2;
    res = -1;

    if (V[med] < med) {
        res = buscar(V, med+1, sup);
    } else if (V[med] == med) {
        res = med;
    }

    return res;
}
```

Ejercicio 2

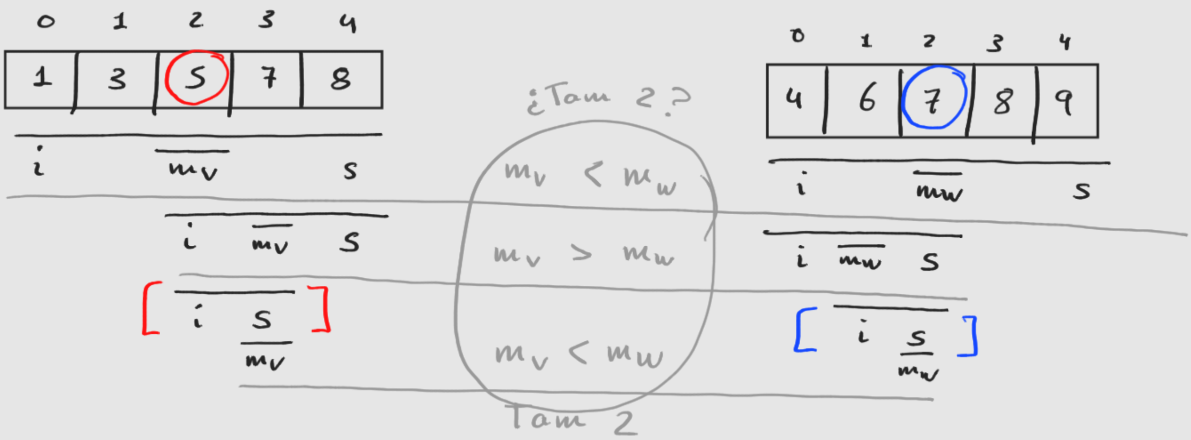
private int buscar (V, inf, sup) {



$2n \longrightarrow n$

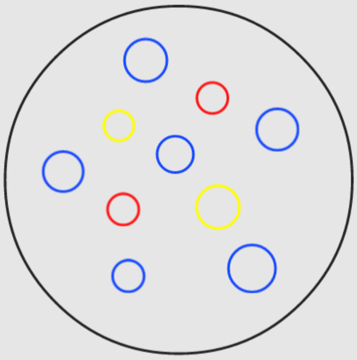
$n \longrightarrow \frac{n}{2}$

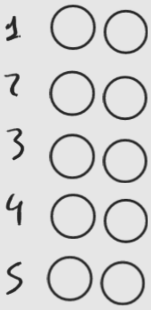
1	3	4	5	5	8	9	10	11	12



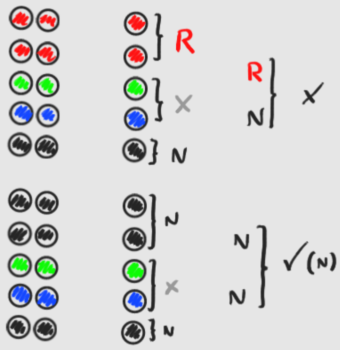
$\min(m_V, m_W)$

Ejercicio 3



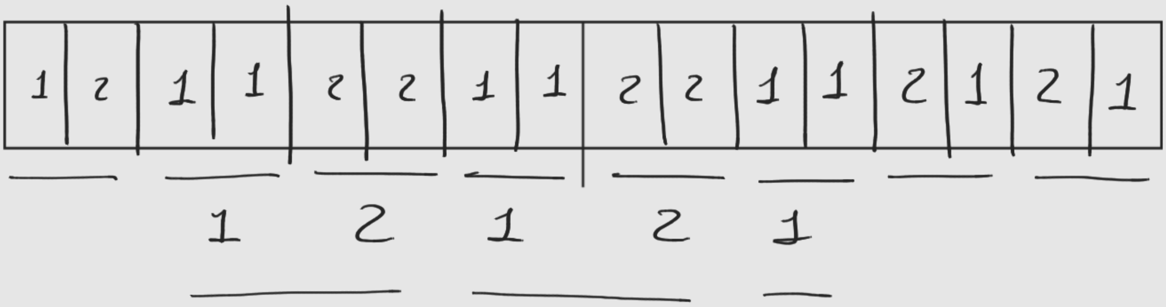


→ Una pareja será

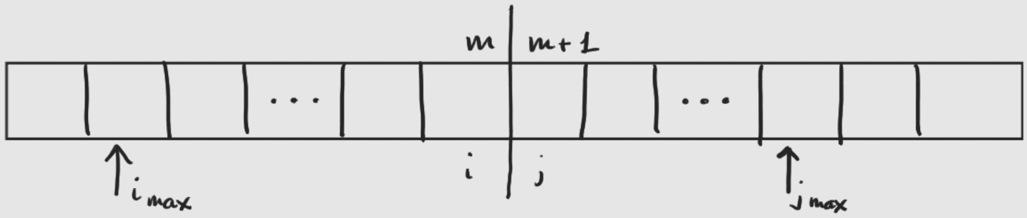


$n = 10$ bolas

$m = 6$ bolas



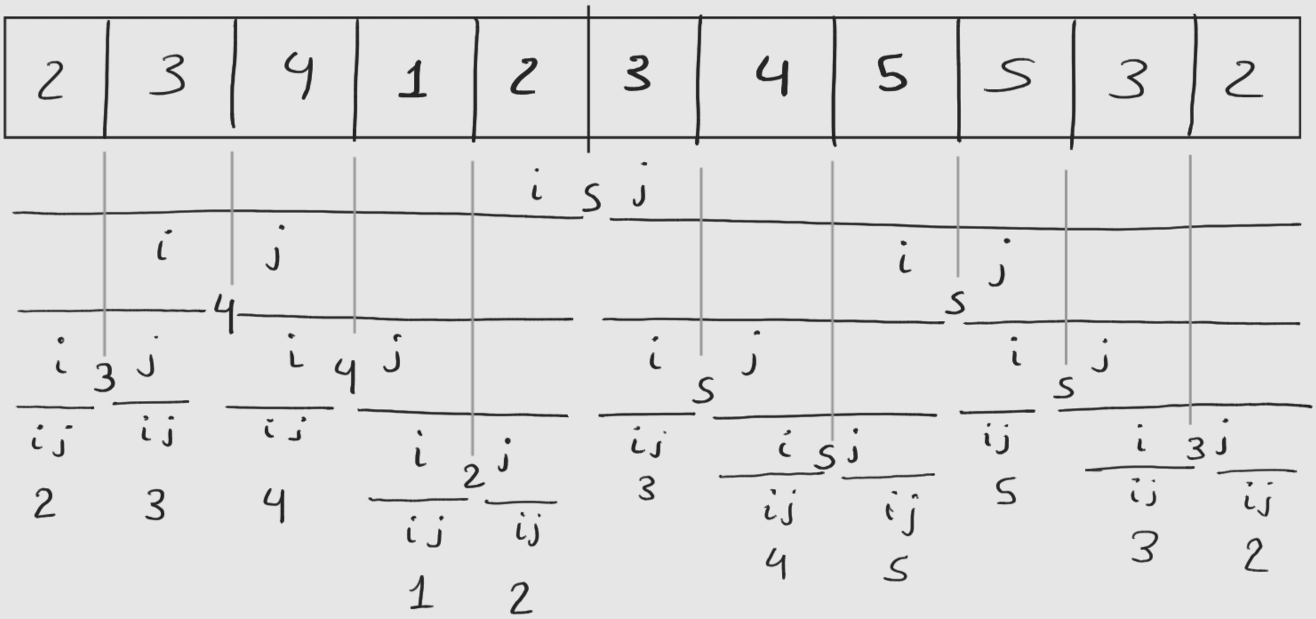
Ejercicio 4



$$m = \frac{(inf + sup)}{2}$$

SumaMAX(inf, sup) :

máximo :



2 3

Ejercicio "3"

Apariciones for n veces: $T_A(n) \in \Theta(n)$.

$$T_{CM}(n) = T_{DP}(n) + 1T_{CM}\left(\tfrac{n}{2}\right) = 1T_{CM}\left(\tfrac{n}{2}\right) + \tfrac{23}{2}n + 1$$

$$T_{DP}(n) = \tfrac{n}{2} + 11n + 1 = \tfrac{23}{2}n + 1$$

$$T_{CM}(n) = 1\,\overset{\Theta(n)}{T\left(\tfrac{n}{2}\right)} + \overset{\Theta(n)\ *}{\tfrac{23}{2}n + 1}$$

$$\begin{cases} a = 1 \\ b = 2 \\ d = 1 \end{cases} \longrightarrow a < b^d \longrightarrow T(n) \in \Theta\left(n^d \cdot \log^k(n)\right)$$

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

$* \ \Theta(n) \subset O(n \log(n))$

Pizarra

Devolver la suma máxima de un vector.

$//pre \equiv \{int[\,]\ V,\ V.length = n,\ n > 0\}$

$//post \equiv \{int\ s : (\exists i \in [0, n-2] : V[i] + V[i+1] = s) \land (\forall j \in [0, n-2] : V[j] + V[j+1] < s) \land (i \neq j)\}$

$//post \equiv \{int\ s : \underbrace{\left(\exists i, j \in [0, n-1],\ i \leq j : \sum_{k=i}^{j}(V[k]) = s\right) \land \left(\forall a, b \in [0, n-1],\ a \leq b : \sum_{t=a}^{b}(V[t]) = s\right)}_{P}\}$

Porque es un subvector, entonces i y j no sabes lo que valen, por eso se usa k, porque «i = 0» es asumir que i empieza en 0 siempre.

$//post \equiv \{int[\,]\ W[i..j] : \ P\} \rightarrow$ devuelve el subvector

$//post \equiv \{int\ i, j : \ P\} \longrightarrow$ devuelve los índices del subvector.

Ejercicio "5"

$//pre \equiv \{int[\,]\ V,\ int\ s,\ *\}$ ($*$: $n \geq 2$)

$//post \equiv \{\exists i, j \in [0, n-1] : (V[i] + V[j] = s,\ i \neq j)\}$

$*$ Precondición: Todos diferentes.

a) $\forall i, j \in [0, n-1] : (i \neq j \rightarrow V[i] \neq V[j])$

b) $\forall i, j \in [0, n-1] : (V[i] = V[j] \longrightarrow i = j)$

Ejercicio [Parcial 19/20]

$//pre \equiv \{int[\,]\ A,\ A.length = n,\ n > 1,\ \forall i \in [0, n-2] : (A[i] < A[i+1],\ A[0] = k,\ k > 0)\}$

$//post \equiv \{int\ m : (\exists i \in [0, n-2] : A[i]+1 \neq A[i+1] \longrightarrow A[i+1] = m) \land (\forall j \in [0, n-1] : A[j]+1 = A[j+1],\ j < i)$

si no: $(A[n-1] + 1 = m)\}$

Completar...

Se define una matriz Kasparoviana:

$$K_0 = \begin{pmatrix} 1 \end{pmatrix} \qquad K_1 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$K_n = \begin{pmatrix} K_{n-1} & -K_{n-1} \\ -K_{n-1} & K_{n-1} \end{pmatrix} \qquad K_2 = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

Se pide diseñar el algoritmo que realice $K_n \times K_n$ por D,V, para que sea más eficiente que $T_{FB}(n) \in \Theta(n^3)$.

Plantear el producto usando K_0, K_1 y K_2.

2^{n-1} elems

$$\underbrace{K_n}_{2^n \text{ elems}} \times K_n = \begin{pmatrix} K_{n-1} & -K_{n-1} \\ -K_{n-1} & K_{n-1} \end{pmatrix} \begin{pmatrix} K_{n-1} & -K_{n-1} \\ -K_{n-1} & K_{n-1} \end{pmatrix} = \begin{pmatrix} 2K_{n-1}^2 & -2K_{n-1}^2 \\ -2K_{n-1}^2 & 2K_{n-1}^2 \end{pmatrix}$$

$$T(2^{n+1}) = T(2^n) + 2^n + 2^n + 2^n + 2^n$$

$$2^{n+1} = k, \quad 2^n = \frac{k}{2} : \quad T(k) = T\left(\frac{k}{2}\right) + 4\frac{k}{2}$$

$$T(k) = T\left(\frac{k}{2}\right) + 2k \quad \in \Theta(k)$$