📝

Ejercicio X

📌 **Especificar formalmente que un array a tenga un único pico.**

Apartado 1

$$//pre\left\{\texttt{a.lenght()} = n > 2\right\}$$

$$//post\left\{p : \exists 1 \leq i \leq n-1,\ (a[i-1] < a[i])\ \wedge\ (a[i] > a[i+1]),\ p = a[i]\right\}$$

$$//post\left\{p : \exists 1 \leq i \leq n-1,\ N_{k\in[1,n-2]}\Big((a[k-1] < a[k])\ \wedge\ (a[k] > a[k+1])\Big) = 1, p = a[i]\right\}$$

En principio esto es con $\wedge$.

Implementación de fuerza bruta.

```
/**
 * Devuelve la posición de un pico en un array de enteros.
 *
 * @param a    Array de enteros con un solo pico
 * @return     El valor del pico
 */
public int devolverPico_FB(int[] a) {
    int i = 1;  // Empieza en 1

    while (a[i-1] < a[i]) {  // Porque empieza creciente
        i++;
    }

    return a[i-1];
}
```

- p: me salgo → $a[p-1] > a[p]$.
- $p-1$: no me salgo → $a[p-2] < a[p-1]$.

Complejidad: $\Theta(n)$.

Vídeo de clase:

```
/**
 * Devuelve la posición de un pico en un array de enteros.
 *
 * @param a    Array de enteros con un solo pico
 * @return     El valor del pico
 */
public int devolverPico_DyV(int[] a, int p, int u) {
    if (p == 1) {
        return(a[p] > a[u] ? a[p] : a[u]);
    }

    int m = (p+u)/2;

    if (a[m-1] < a[m] && a[m] > a[m+1]) {
        return a[m];

    } if (a[m-1] > a[m]) {
        return devolverPico_DyV(a, p, m);
    }
```

```
    return devolverPico_DyV(a, m, u);
}
```

$$\texttt{if (a[m-1] < a[m] \&\& a[m] > a[m+1])}$$

$$\neg\Big(a[m-1] < a[m] \quad \text{AND} \quad a[m] > a[m+1]\Big)$$

$$\neg\Big(a[m-1] < a[m]\Big) \quad \text{OR} \quad \neg\Big(a[m] > a[m+1]\Big)$$

$$a[m-1] \geq a[m] \quad \text{OR} \quad a[m] \leq a[m+1])$$

📝 En el examen, la información redundante no afectará a la puntuación. Es decir, el código no debe ser el más óptimo, pero sí cumplir con la especificación.

$$T_{\texttt{devolverPico\_DyV}}(n) = T_{\texttt{devolverPico\_DyV}}\left(\frac{n}{2}\right) + k$$

$$\begin{cases} a & = 1 \\ b & = 2 \\ d & = 0 \quad \to k \cdot n^0 \end{cases} \quad \longrightarrow \quad T_{\texttt{devolverPico\_DyV}}(n) \in \Theta\Big(n^d \cdot \log(n)\Big)$$

$$T_{\texttt{devolverPico\_DyV}}(n) \in \Theta\Big(\log(n)\Big)$$