Ejercicio 1

a)

$$T(0) = 6$$

$$T(1) = \frac{5}{2} + \frac{5}{2} + 6 = 5 + 6 = 11$$

$$T(2) = \frac{5 \cdot 4}{2} + \frac{5 \cdot 2}{2} + 6 = 10 + 5 + 6 = 21$$

$$T(0) = 6$$
$$T(1) = 11$$
$$T(2) = 21$$

$$T(n) = 7T(n-2) - 6T(n-3)$$

$$T(n) + 0T(n-1) - 7T(n-2) + 6T(n-3) = 0$$

$$x^3 + 0x^2 - 7x^1 + 6x^0 = 0$$

$$x^3 - 7x + 6 = 0$$

	1	0	-7	6
1		1	1	-6
	1	1	-6	0
2		2	6	
	1	3	0	
-3		-3		
	1	0		

$\rightarrow$ $x^3 - 7x + 6 = 0$

$(x-1)(x-2)(x+3) = 0$

$$\begin{cases} r_1 = 1 & m_1 = 1^n \\ r_2 = 2 & m_2 = 2^n \\ r_3 = -3 & m_3 = (-3)^n \end{cases}$$

Solución parcial: $T(n) = a + b \cdot 2^n + c \cdot (-3)^n$

[1] $\begin{cases} T(0) = a + b + c \\ T(0) = 6 \end{cases}$

[2] $\begin{cases} T(1) = a + 2b - 3c \\ T(1) = 11 \end{cases}$

[3] $\begin{cases} T(2) = a + 4b + 9c \\ T(2) = 21 \end{cases}$

$$\begin{cases} a + b + c = 6 & [1] \\ a + 2b - 3c = 11 & [2] \\ a + 4b + 9c = 21 & [3] \end{cases}$$

[1] $a = 6 - b - c$

$$\begin{cases} (6-b-c)+2b-3c=11 \\ (6-b-c)+4b+9c=21 \end{cases} \rightarrow \begin{cases} b-4c=5 & [A] \\ 3b+8c=15 & [B] \end{cases}$$

$[A]\quad b = 4c+5$

$[B]\quad 3(4c+5)+8c = 15$

$12c+15+8c=15$

$20c = 0$

$c = 0$

$[A]\quad b = 4(0)+5$

$b = 5$

$[1]\quad a = 6-(5)-(0)$

$a = 1$

$$(a,b,c) = (1,5,0)$$

$$T(n) = (1) + (5)\cdot 2^n + (0)\cdot 3^n$$

$$T(n) = 5\cdot 2^n + 1$$

$$T(n) \in \Theta(2^n)$$

b)

$$T(n) \in \begin{cases} a < b^d \rightarrow O(n^{\log_b(a)}) \\ a = b^d \rightarrow O(n^d \cdot \log(n)) \\ a > b^d \rightarrow O(n^d) \end{cases}$$

a) $a=4,\ b=2,\ d=1$ $\quad T(n) = 4T(\frac{n}{2}) + n$

b) $a=2,\ b=2,\ d=1$ $\quad T(n) = 2T(\frac{n}{2}) + n$

c) $a=4,\ b=2,\ d=2$ $\quad T(n) = 4T(\frac{n}{2}) + n^2$

c)

$f_1(n) = \log(\log(n))$ $\qquad f_2(n) = (\log(n))^2$

Ejercicio 2

$0 \in \mathbb{Z}^+$

//pre $\{$ int[] V, V.length $= n$, $n > 0$, $i \in [0, n-1] : V[i] > 0 \}$

//post $\{$ int $x : \left(i \in [0, n-1] : \bigwedge_{i=0}^{n-1} \left(V[i] \mathbin{\%} 2 \neq 0 \right) = x \right) \}$

Ejercicio 3

Completar

Ejercicio 4

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

//post $\{$int f : $\left(\forall i \in [0, n-2], \forall j \in [i, n-2] : (A[j+1] \neq A[j]+1) \wedge \neg(A[i+1] \neq A[i]+1,\ i<j) \Rightarrow A[j] = f\right)$

$\vee$ (A.length $= 1 \Rightarrow A[0]+1 = f)\}$

$A_1 = \{1\} \rightarrow 2$

$A_2 = \{1, \underbrace{2, 4}_{3}\} \rightarrow 3$

$A_3 = \{1, 2, 3, 4\} \rightarrow 5$

$A_4 = \{1, \underbrace{2, 4}_{3}, \underbrace{5, 7}_{6}, 8\} \rightarrow 3$

Corregir

b)

```
private int menorSuperior(int[] A) {
    if (A.length == 1) {
        return A[0] + 1;

    } else {
        int i = 0;
        while(A[i] + 1 != A[i + 1] && i < A.length - 1) {
            i++;
        }

        return A[i] + 1;
    }
}
```

c) Complejidad n porque es un bucle y en en peor de los casos, lo recorre entero.

Como está en un if y su complejidad es la mayor del if o el else, se mantiene para n en el peor caso.

$$\text{menorSuperior(int[] V)} \in O(n)$$

d)

```
public int menorSuperior(int[] A) {
    if (A.length == 1) {
        return A[0] + 1;
    } else {
        return buscar(A, i, A.length);
    }
}


private int buscar(int[] A, int i, int s) {
    if (i == s) {
        if (A[i] - 1 == A[0]) {
            min = A[i] + 1;
        } else {
            min = A[i-1] + 1;
        }
    } else {
        int m = (i + s) / 2;
        if (A[m] - m = A[0]) {
            min = buscar(A, i-1, s);
        } else {
            min = buscar(A, i, s-1);
        }
    }
}
```

e)

$$T(n) = 1T\left(\frac{n}{2}\right) + cte, \qquad cte \in \mathbb{R}$$

$$\begin{cases} a = 1 \\ b = 2 \\ d = 0 \end{cases} \longrightarrow a = b^d \rightarrow T(n) \in \Theta\left(n^d \cdot \log(n)\right)$$

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

$S(n) =$ ¿