TA1 Equipo 1

Tipo	Tarea
Estado	Terminado 👌
Fecha	@March 29, 2022
Orden	5
Related to Recordatorios (Property)	
Related to Asignaturas (Tareas)	Algoritmos
Property	

Ejercicio 1

```
public static int enRango (int[] a, int bajo, int alto) {
  int contador = 0; O(1)
  for (int i=0; i<a.length; i++) condicion: O(1) Bucle: { O(a.length) -> O(n)
    if (a[i] >= bajo && a[i] < alto) O(1)
    contador++; O(1)
  }
  return contador; O(n)
}
Total:
   Es de orden O(n)
```

Ejercicio 2

```
unaFunción ( N de tipo entero)
 i <- 1 O(1)
 j <- N O(1)
 mientras i < N hacer O(1)
 j <- N – 1 O(1)
 i <- i * 2 O(1)
 fin mientras
 devolver (j) O(1)
fin

Total:
       O(Log N) porque se va multi la i
```

Ejercicio 3

```
int[] cuentas = new int [100]; O(1)
for (int i = 0; i<100; i++) {   O(100)
 cuentas[i] = enRango (notas, i, i+1); O(n)
}

Total: es O(100) * O(n) en total es 0(n)
```

Ejercicio 4

```
unValor (A, N de tipos enteros)
i <- 0
Si N < 3 entonces O1
devolver (A)
fin si
mientras i < 3 hacer O(3)
  si arreglo[i] = A entonces O(1)
    devolver ((arreglo[0] + arreglo[N-1]) div 2) O(1)
  fin si
  i <- i + 1 O(1)
fin mientras
devolver (A div N) O(1)
Fin

Total:
  es O(1)
```

Ejercicio 5

```
otraFunción (claveAbuscar)
  inicio <- 0 O(1)
  fin <- N-1 O(1)
    mientras inicio ≤ fin hacer O(n-1), condicion O(1)
     medio <- (inicio + fin) div 2 O(1)
       si (arreglo[medio] < claveAbuscar) entonces O(1)
         inicio <- medio + 1 O(1)
         sino
           si (arreglo[medio] > claveAbuscar) entonces O(1)
             fin <- medio – 1 O(1)
           sino O(1)
         devolver medio
       fin si
     fin si
    fin mientras
  devolver -1 O(1)
fin

Total:
 O(Log n) vas descartando de mitad en mitad
```

Ejercicio 6

```
function particion( i, j: integer; pivote: TipoClave): integer;
  {divide V[i], .., V[j] para que las claves menores que pivote estén a la
  izquierda y las mayores o iguales a la derecha. Devuelve el lugar donde se
  inicia el grupo de la derecha.}
  COMIENZO
     L <- i; O(1)
     R <- j; O(1)
   Repetir  total del Bucle
     intercambia(V[L],V[R]); O(1)
     mientras V[L].clave < pivote hacer L := L + 1; fin mientras O(1)
     mientras V[R].clave >= pivote hacer R := R - 1; fin mientras O(1)
   Hasta que L > R O()
   Devolver L; O(1)
FIN; {particion}

Total: O(j-i)
   Casos
```

Ejercicio 7

```
miFunción
  Desde i = 1 hasta N-1 hacer  (Se repite n veces) O(n-1)
   Desde j = N hasta i+1 hacer (Se repite n veces) Condicion O(1), Bucle O(n) -> se repite n-2 veces
     Si arreglo[j].clave < arreglo[j-1].clave entonces O(1)
       Intercambia(arreglo[j], arreglo[j-1]) O(1)
     Fin si
   Fin desde
  Fin desde
Fin

Total: O(n^2)
```