

 <small>Universidad Autónoma de Occidente</small>	UNIVERSIDAD AUTÓNOMA DE OCCIDENTE				 <i>Valoración</i>
	FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS		ESTRUCTURAS DE DATOS Y ALGORITMOS 1	GRUPO	
	CÓDIGO:		NOMBRE:		
TALLER ESTUDIO CORTE 2					FECHA:

PRIMERA PARTE	Calculo del número total de instrucciones ejecutadas		
---------------	--	--	--

Para cada uno de los siguientes algoritmos: determinar de qué tipo de dato son los parámetros, interpretar cuando se da el peor caso y calcular el número total de instrucciones ejecutadas.

<div>a)</div> <pre>function item1(A){ let res = "Si se cumple"; let final = A.length-1; let mitad = A.length/2; for(let i=0; i<mitad; i++){ if (A[i] > A[final]){ res = "no se cumple"; break; } final--; } return res; }</pre>	<div>b)</div> <pre>function item2(n){ let tope = Math.trunc(Math.sqrt(n)); let res = true; for(let i=2; i<=tope; i++){ if (n%i==0){ res =false; break; } } return res; }</pre>
---	--

c)

```
function item3(A){
  let cont=0;
  let suma = 0;
  for (let i=2; i<A.length-2; i++){
    for(let j=0; j<=A.length/2; j++){
      if (A[j]%2==0){
        cont++;
      }else{
        suma += A[j];
        A[j]+=2;
      }
    }
  }
  return cont*suma;
}
```

d)

```
function item4(A,p,q){
  let veces=0;
  let acum = 0;

  for (let i=p; i<=q; i++){
    if (A[p] < 50000){
      veces++;
      acum+=A[p];
      A[p]+=2000;
    }else{
      n-2 ⇒ A[p] +=item5(A,A[p])
    }
  }
  return veces+ " " + acum;
}

console.log(item4(info,0,info.length-1));
```

En el que se indica el numero de instrucciones ejecutadas por la linea en la que aparece la flecha

Para cada uno de los siguientes algoritmos: determinar de qué tipo de dato son los parámetros, interpretar cuando se da el peor caso y calcular su complejidad O grande.

a) Para cada uno de los casos de la primera parte

b) de los métodos Java BubbleSort, printArray y main, existentes en <https://www.geeksforgeeks.org/bubble-sort/>

c) de las funciones JavaScript insertionSort, printArray y driver code, existentes en <https://www.geeksforgeeks.org/insertion-sort/>

d)

```
function item5(A){
  for (let i=1; i<A.length-1; i++){
    for (let j=0; j<Math.trunc(Math.log2(A.length)); j++){
      console.log(A[i]+' '+A[j]);
    }
  }
}
```

e)

```
INCOGNITA_SEARCH(A,p,q,r)
  m = (p+q)/2
  Mientras p<=q y A[m] diferente r
    Si r < A[m]
      q = m-1
    Sino
      p = m+1
  m = (p+q)/2
  FinMientras
  posicion = -1
  Si r igual a A[m]
    posicion = m
  Devolver posicion

INCOGNITA_SEARCH( info, 0, info.length-1, dato )
```

f)

```
function item6(datos){
  let hasta = Math.trunc(datos.length)
  for (let i=0; i<=hasta ; i++){
    if (datos[i]>datos[hasta]){
      item5(datos)
    }else{
      INCOGNITA_SEARCH(datos,0,datos.length-1,A[i])
    }
  }
}
```