



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Tarea 3

Introducción a Ciencias de la Computación

Rodrigo André Decuir Fuentes



Teoría

Responde las siguientes preguntas:

1. ¿Qué son los arreglos y para qué son útiles?

- Los arreglos son conjuntos finitos y ordenados de elementos homogéneos, asimismo son objetos.
- Son de mucha utilidad para manejar de forma sencilla y directa un conjunto de datos del mismo tipo.

2. Se tiene el siguiente arreglo:

2	35	-22	0	56	78	-84	8	43	1
---	----	-----	---	----	----	-----	---	----	---

Ordena el arreglo anterior por medio de

- Bubble Sort
- Selection Sort

Comportamiento Bubble Sort

Comportamiento Selection Sort

3. ¿Qué es la recursión y en qué se aplica? ¿Qué componentes debe tener un método recursivo?

- La recursión es la forma de expresar la solución a un problema en términos de sí mismo, es aplicada para trabajar en cosas que tienen muchas ramificaciones posibles y son demasiado complejas para un enfoque iterativo. Un buen ejemplo de esto sería buscar a través de un sistema de archivos.
- Toda función recursiva tiene dos componentes: un caso base y un paso recursivo.

4. Se tienen los siguientes dos programas:

```

1
2 public double mist (int n) {
3     double s = 0.0;
4     for (double i = 1; i <= n ; i ++ ) {
5         s = s + 1/ i ;
6     }
7     return s ;
8 }
9

```

Listing 1: mist

```

1
2 public double mistR (int n) {
3     if (n <= 1) {
4         return 1;
5     } else {
6         return ((double) 1/ n) + mistR (n - 1) ;
7     }
8 }
9

```

Listing 2: mistR

¿Qué hacen mist y mistR?

- **mist**: recibe un entero,
- **mistR**: recibe un entero,

¿Cuál de las dos versiones es mejor y por qué?

La mejor versión es **mist** , ya que...

5. ¿Por que es tan importante la Complejidad en Ciencias de la Computación?

Es importante porque nos permite determinar la eficacia de un algoritmo. La cual se expresa en función del tamaño de la entrada. Para poder representar la complejidad se hace uso de la notación O grandota la cual permite evaluar el ritmo de crecimiento en el tiempo de ejecución como una función dependiente de la cantidad de datos.

6. Se tiene el siguiente programa:

```

1
2 public int mist (int n) {
3     int r ;
4     for (r = 0; n != 0; n /= 10) {
5         r += n % 10;
6     }
7     return r ;
8 }
9

```

Listing 3: mist

```

1
2 public int mistR (int n) {
3     if (n == 0) {

```

```

4         return 0;
5     } else {
6         return (n % 10) + mistR (n / 10) ;
7     }
8 }
9

```

Listing 4: mistR

¿Cuál es la complejidad de mist y mistR? Justifica tu respuesta.

- **mist:**
- **mistR:**

¿Cuál método utiliza la regla de suma y cuál la regla de la multiplicación?