

TAREA 2: ESTRUCTURAS DE DATOS Y ALGORITMOS 2^{do} Semestre 2020**Profesores:** Javier Carrión, Miguel Rodríguez**Ayudantes:** Nicolás Nuñez, Yerko Ortiz, Alan Toro**Objetivo de la tarea:** Evaluar análisis de algoritmos usando notación Big-Oh $O(f(n))$.

Para la presente tarea usted deberá crear un informe de manera individual, en el cuál deben estar sus respuestas a las siguientes preguntas.

Análisis de la tarea 1

A continuación se le entregarán distintos códigos solución de la tarea 1. Su tarea consiste en analizar los tiempos de complejidad de cada uno, junto con una breve descripción del funcionamiento de estos mismos. Adicionalmente, deberá comparar su solución implementada para la tarea 1 respecto los códigos entregados, para esto es importante que en el informe esté su código, como así mismo indique cuál de todos los códigos es el más eficiente en términos de tiempo.

Número faltante

Solución 1

```
1 static int solution1(int[] arr, int n) {
2     int sum_n = (n * n + n) / 2;
3     for(int i = 0; i < n - 1; ++i)
4         sum_n -= arr[i];
5     return sum_n;
6 }
```

Solución 2

```
1 static int solution2(int[] arr, int n) {
2     boolean flag = false;
3     int num_faltante = 0;
4     for(int i = 1; i < n + 1; ++i) {
5         for(int j = 0; j < n - 1; ++j) {
6             if(arr[j] == i){
7                 flag = true;
8                 break;
9             }
10        }
11        if(flag == false) {
12            num_faltante = i;
13            break;
14        }
15        flag = false;
16    }
17    return num_faltante;
18 }
```

Solución 3

```
1 static int solution3(int[] arr, int n) {
2     boolean[] table = new boolean[n];
3     int number = 0;
4     for(int i = 0; i < n - 1; ++i)
5         table[arr[i] - 1] = true;
6     for(int i = 0; i < n; ++i)
7         if(table[i] == false)
8             number = i + 1;
9     return number;
10 }
```

Daño recibido por el mago (El mago y los dragones)

Solución 1

```
1 static int solution1(int ps[], int dps[], int n) {
2     int sum = 0,
3     dps_sum = 0;
4     for(int i = 0; i < n; ++i)
5         dps_sum += dps[i];
6     for(int i = 0; i < n; ++i) {
7         sum += (dps_sum * ps[i]);
8         dps_sum -= dps[i];
9     }
10    return sum;
11 }
```

Solución 2

```
1 static int solution2(int ps[], int dps[], int n) {
2     int sum = 0;
3     for(int i = 0; i < n; ++i)
4         for(int j = i; j < n; ++j)
5             sum += (dps[j] * ps[i]);
6     return sum;
7 }
```

Unos o ceros

Solución 1

```
1 static void solution(char[] str, int index){
2     if(index == str.length){
3         System.out.println(str);
4         return;
5     }
6     if(str[index] == '?'){
7         str[index] = '0';
8         recursive(str, index + 1);
9         str[index] = '1';
10        recursive(str, index + 1);
11        str[index] = '?';
12    }
13    else{
14        recursive(str, index + 1);
15    }
16 }
```

Análisis de algoritmos

Dados los siguientes códigos, su tarea será identificar la complejidad en tiempo para cada uno de los algoritmos. Adicionalmente debe realizar una breve descripción de cada uno, explicando lo que los códigos hacen con sus propias palabras.

Algoritmo a

```
1 static void algo(int arr[]) {
2     int n = arr.length;
3     for (int i = 0; i < n-1; i++) {
4         int min_idx = i;
5         for (int j = i+1; j < n; j++)
6             if (arr[j] < arr[min_idx]) min_idx = j;
7         int temp = arr[min_idx];
8         arr[min_idx] = arr[i];
9         arr[i] = temp;
10    }
11 }
```

Algoritmo b

```
1 static void algo(int[] array, int n){
2     int i, j, size_table;
3     int max = ~0, min = ~(~0);
4     for(i = 0; i < n; ++i) {
5         if(max < array[i]) max = array[i];
6         if(min > array[i]) min = array[i];
7     }
8     size_table = max - min + 1;
9     int[] table = new int[size_table];
10    for(i = 0; i < n; ++i)
11        ++table[array[i] - min];
12    for(i = 0, j = 0; i < size_table; ++i){
13        if((table[i]--) > 0) array[j] = (i + min);
14        else ++j;
15    }
16 }
```

Algoritmo c

```
1 static int algo(int a, int b){
2     if(b == 0) return a;
3     return algo(b, a % b);
4 }
```

Consideraciones

- La fecha de entrega de esta tarea será para el día miércoles 2 de septiembre a las 07:00 AM.
- Todo envío que se realice después de la fecha de entrega será evaluado con nota 1.0.
- Para esta tarea en particular usted ha de realizar un informe en formato PDF respondiendo cada una de las preguntas del enunciado, este informe debe ser enviado vía Canvas respetando la fecha de entrega.
- Para cualquier consulta sobre la tarea consulte a su ayudante correspondiente: nicolas.nunez2@mail.udp.cl, yerko.ortizm@mail.udp.cl, alan.toro@mail.udp.cl.
- La acción de plagio o copia será penalizada con la reprobación del ramo.