ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 2: LABORATORY #2

Alejandro Torres Muñoz
Universidad Eafit
Medellín, Colombia
eatorresm@eafit.edu.co

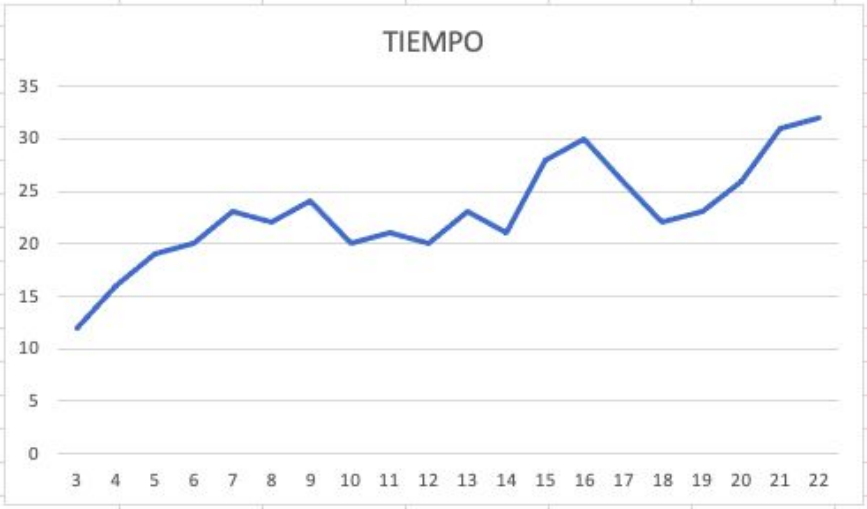
Mateo Muñoz Cadavid
Universidad Eafit
Medellín, Colombia
mmunozc4@eafit.edu.co

3) Practice for final project defense presentation

3.1 [OPTIONAL] , 3.2

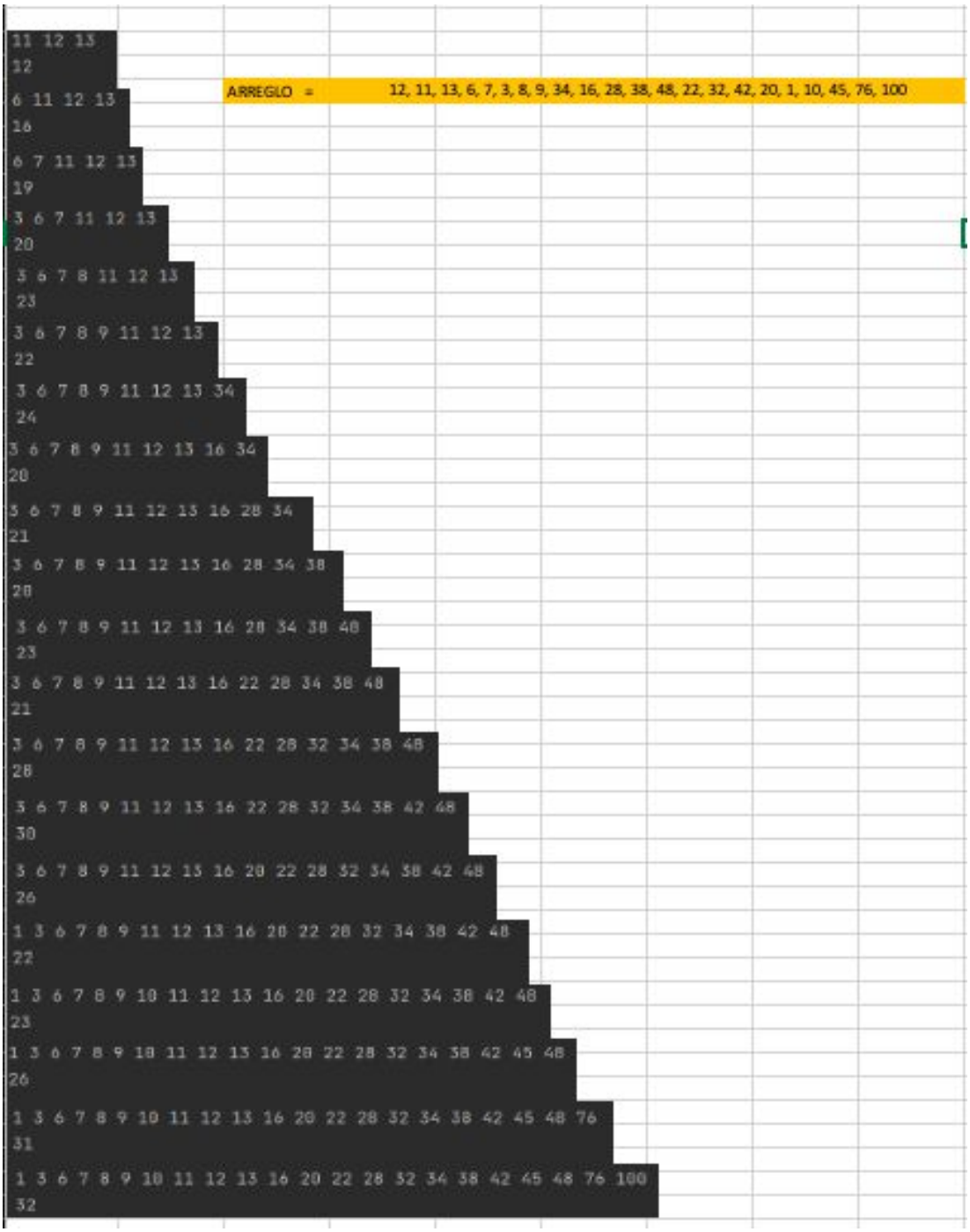
Insertion Sort

TAMAÑOS	TIEMPO
3	12
4	16
5	19
6	20
7	23
8	22
9	24
10	20
11	21
12	20
13	23
14	21
15	28
16	30
17	26
18	22
19	23
20	26
21	31
22	32



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

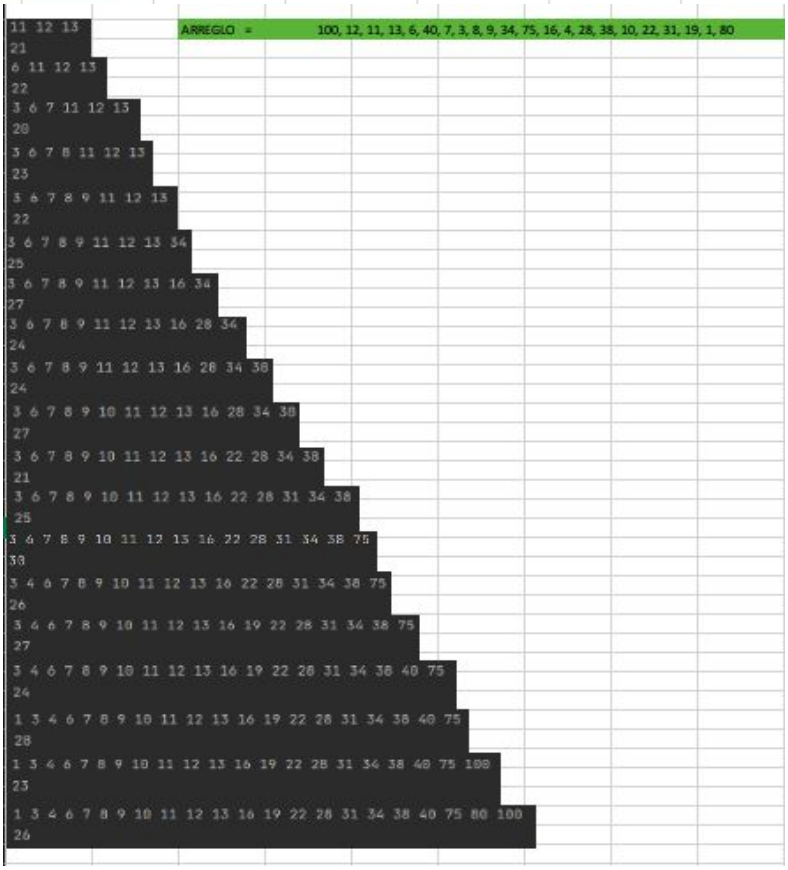
ESTRUCTURA DE DATOS 1
Código ST0245



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

Merge Sort

TAMAÑOS	TIEMPO
3	21
4	22
5	20
6	20
7	23
8	22
9	25
10	27
11	24
12	24
13	27
14	21
15	25
16	30
17	26
18	27
19	24
20	28
21	23
22	26



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3.3 It is not appropriate due to its algorithmic complexity for crafting a game with millions of images, elements in a scene, and real-time demands on rendering 3D scenes. It would be many operations to do, therefore, it is not efficient.

3.4 The logarithmic complexity of the MergeSort is due to the fact that it divides the arrangement into arrangements of two arrangements each time, therefore, to arrive at the arrangement of an element it must perform logn operations, subsequently to combine all these arrangements in a combined way it must perform n operations and hence the multiplication of this complexity gives nlogn.

3.4 [OPTIONAL]

MaxSpan begins by traversing the array to divide it into two positions, "i" and "j", and then compares whether the value found in those two positions of the array is equal. If it is equal, the variable cont will be equivalent to the value of the subtraction between i and j plus 1, and the value of varMax will correspond to the calculation of the maximum value between the corresponding value in cont and the value of varMax (initially at 0). This will be repeated until the algorithm goes through the entire array to finally return the final value of varMax.

3.5 [OPTIONAL]

Merge Sort is more effective in terms of using a large amount of data and elements, while Insertion Sort uses a smaller and reduced number of elements in it. Due to this, in large size arrangements it is proposed that the data be ordered or equal, in this way we will make Insertion Sort even more efficient and faster than Merge Sort.

In shorter times n ^ 2 (Insertion Sort) is faster than nlogn (Merge Sort)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3.5, 3.6

Array 2:

1. Sum13:

```java
public int sum13(int[] nums) {
        int sum = 0;
        int i = 0;
        if(nums.length==0){
            return 0;
        }
        while(i<nums.length){
            if(nums[i]==13){
                i+=2;
            }else{
                sum+=nums[i];
                i++;
            }
        }
        return sum;
    }
//O(n)
//n corresponds to the number of elements in the array nums
```

2. modThree

```java
public boolean modThree(int[] nums) {
  if(nums.length <3){
    return false;
  }
  for(int i = 0; i<nums.length-2; i++){
    if((nums[i] % 2 == 0) && (nums[i+1] % 2 ==0) && (nums[i+2] %2==0)){
      return true;
    }else if((nums[i] % 2 != 0) && (nums[i+1] % 2 !=0) && (nums[i+2] %2!=0)){
      return true;
    }
  }
  return false;
}

//O(n)
//n corresponds to the number of elements of the array to be evaluated.
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3. only14

```java
public boolean only14(int[] nums) {
        for(int i=0; i<nums.length;i++){
            if((nums[i]!=1) && (nums[i]!=4)){
                return false;
            }
        }
        return true;
    }
//O(n)
//n length nums
```

4. sum28

```java
public boolean sum28(int[] nums) {
  int totalSumArray = 0;
        for (int i = 0; i <nums.length ; i++ ) {
            if(nums[i] == 2){
              totalSumArray +=2;
            }
        }
        if(totalSumArray == 8){
          return true;
        }
        return false;
    }
//O(n)
//n corresponds to the number of elements in the array nums
```

5. centeredAverage

```java
public int centeredAverage(int[] nums) {
        int cont = 0;
        int vMax = nums[0];
        int vMin = nums[0];

        for(int i=0; i<nums.length;i++){
            cont+=nums[i];
            vMin = Math.min(vMin, nums[i]);
            vMax = Math.max(vMax, nums[i]);
        }
        return (cont-vMax-vMin)/(nums.length-2);
    }
//O(n)
// n corresponde a la longitud del arreglo
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

Array 3:

1. maxSpan

```java
public int maxSpan(int[] nums) {
 int cont = 0;
 int varMax = 0;

 for(int i = 0; i<nums.length; i++){
   for(int j = 0; j<nums.length; j++){
     if(nums[i] == nums[j]){
       cont = j-i+1;
       varMax = Math.max(cont, varMax);
     }
   }
 }
 return varMax;
}
//O(n)
// n is a length of the array.
```

2. canBalance

```java
public boolean canBalance(int[] nums) {

  for(int i = 0; i<nums.length; i++){
    int first = 0;
    int second = 0;
    for(int j = 0; j<=i; j++){
      first += nums[j];
    }
    for(int k = i+1; k<nums.length; k++){
      second += nums[k];
    }
    if (first == second){
      return true;
    }
  }
  return false;
}
//O(n)
//n corresponds to the number of elementos in an array of nums.
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

3. CountClumps

```java
public int countClumps(int[] nums) {
        int cont= 0;
        boolean clump = false;
        for(int i = 0; i < nums.length - 1; i++)
        {
            if(clump)
            {
                if(nums[i] != nums[i+1])
                    clump = false;
            }
            else if(nums[i] == nums[i+1])
            {
                clump = true;
                cont++;
            }
        }
        return cont;
    }
}
//O(n)
//n is the number of values in the array
```

4. seriesUp

```java
public int[] seriesUp(int n) {
        int[] arr = new int[n*(n+1)/2];
        int pos = 0;
        for(int i = 1; i <= n; i++){
            for(int j = 1; j <= i; j++, pos++){
                arr[pos] = j;
            }
        }
        return arr;
}
//O(n^2)
//n is the number of times to repeat the pattern
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

5. squareUp

```
    public int[] squareUp(int n){
        int[] arr = new int[n*n];
        int pos;
        for(int i = 1; i <= n; i++){
            pos = n * i - 1;
            for(int j = 1; j <= i; j++, pos--){
                arr[pos] = j;
            }
        }
        return arr;
}

//O(n^2)
//n length nums
```

4) Practice for midterms

4.1 The time this algorithm will take is 1000 ms.
4.2 *b) O(m×n×√n)*
4.3 *a) O(n^3)`*
4.4 *1. O(m+n+m*n)*
*2. O(n*m)*
4.5 *1. d) T(n)=T(n/10)+c, which is O(log10 n)*
2. Yes

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

6) Team work and gradual progress (optional)

Member	Date	Done	Do	To do
Alejandro Torres	4/03/21	I read the lab guide		Inquire about the Insertion Sort and Merge Sort algorithms.
Mateo Muñoz	4/03/21	I read the lab guide		Investigate the complexity of algorithms.
Alejandro Torres	4/03/21	I inquired about the Insertion Sort and Merge Sort algorithms.	Review research of Mateo.	Establish the development environment and implement both algorithms.
Mateo Muñoz	4/03/21	I did research on the complexity of algorithms.	Watch a video about Insertion Sort and Merge Sort..	Establish the programming language to use and implement for both algorithms.
Alejandro Torres	4/03/21	I set the development environment	Develop the algorithm.	Test the algorithm
Mateo Muñoz	8/03/21	I established the programming language.	Develop the algorithm.	Solve exercises online.

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

Alejandro Torres	8/03/21	I tested the algorithm.		Solve exercises online.
Mateo Muñoz	9/03/21	I solved exercises online (Array 2, 3).	Analyze the exercises that Alejandro solved.	Analyze the complexity of the online exercises.
Alejandro Torres	9/03/21		Analyze the exercises that Mateo solved.	Analyze the complexity of the online exercises.
Mateo Muñoz	9/03/21	I analyzed the complexity of the online exercises (Array 2,3).	Detail the complexity of the exercises that Alejandro did.	Analyze the execution times of the Insertion Sort and Merge Sort algorithms.
Alejandro Torres	9/03/21		Detail the complexity of the exercises that Mateo did.	Analyze the feasibility of the Insertion Sort algorithm for a 3D video game.
Mateo Muñoz	11/03/21	I analyzed the execution times of the algorithms.		Generate a graph of the times each algorithm took and analyze.
Alejandro Torres	11/03/21	I analyzed and answered about the feasibility of the algorithm.		
Mateo Muñoz	11/03/21	We generate a graph about the times.	Check Alejandro's answer.	Analyze the results of the graph.

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

Alejandro Torres	11/03/21		Check Mateo's answer.	Explain how maxSpan works.
Mateo Muñoz	11/03/21	I analyzed the results of the graph.		Build a table for both algorithms, and analyze how long it would take for a size of 20.
Alejandro Torres	11/03/21			
Mateo Muñoz	11/03/21	I analyzed the results of the table.		Explain the variables of the complexity of the online exercises.
Alejandro Torres	11/03/21	I explained how maxSpan works.		
Mateo Muñoz	11/03/21	We explain the complexity of the exercises.	I analyzed Alejandro's answer.	Begin exam mock exercises.
Alejandro Torres	11/03/21		I analyzed the results obtained by Mateo.	
Mateo Muñoz	12/03/21	We finished the mock exam exercises.		Answer on asymptotic complexity for the worst case, of merge sort or insertion sort.

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

Alejandro Torres	12/03/21			Answer about how the data should be so that insertion sort is faster than merge sort.
Mateo Muñoz	12/03/21	I answered about the asymptotic complexity.		Design a template for gradual progress.
Alejandro Torres	12/03/21	I answered about the insertion sort and merge sort data.		
Mateo Muñoz	13/03/21	We design a template for gradual progress.		Finalize delivery details.
Alejandro Torres	13/03/21			
Mateo Muñoz	14/03/21	We finalize delivery details.		DONE.
Alejandro Torres	14/03/21			

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473