ESTRUCTURA DE DATOS 1
Código ST0245

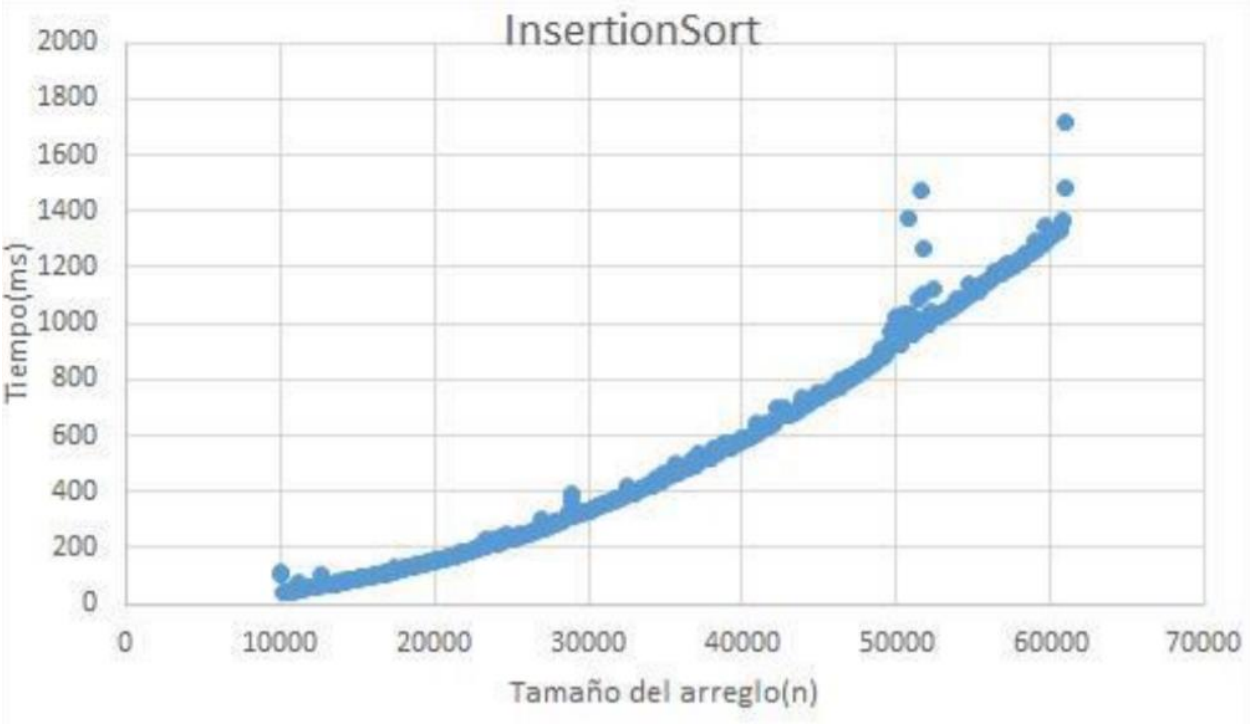
Laboratorio Nro. 2
Complejidad de Algoritmos

Juan Pablo Henao Bedoya
Universidad Eafit
Medellín, Colombia
jphenaob@eafit.edu.co

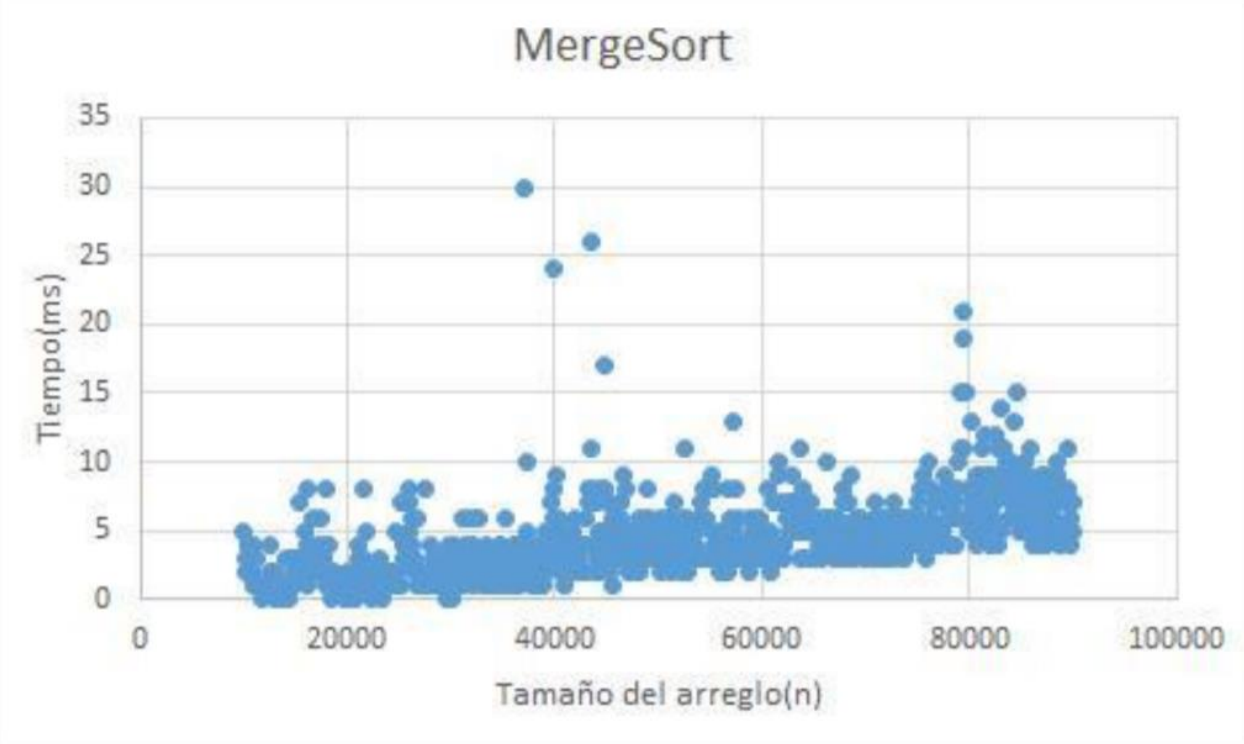
Diego Alejandro Vanegas González
Universidad Eafit
Medellín, Colombia
davanegasg@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.2



PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



3.3 No es lo mas adecuado, ya que un videojuego tiene gran cantidad de datos, y el tiempo que toma el insertionSort es muy alto.

3.4 Debido a que el MergeSort tiene O(n log n) de complejidad por su eficiencia tanto en tiempo como en orden

3.5 [Opcional] Generalmente deben ser datos ordenados, y simples, como, por ejemplo, un ordenamiento de números.

3.5

```
public class TripleUp
{
  public boolean tripleUp(int[] nums) {
  boolean siono = false;                              O(1)
  int n = 0;                                          O(1)
  for(int i = 0; i < nums.length-2; i++){             O(n)
    n=nums[i];                                        O(1)
    if(nums[i+1]==n+1 && nums[i+2]==n+2){             O(1)
      siono=true;                                     O(1)
    }
  }
  return siono;                                       O(1)
}
```

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

```
    }          O(1)+O(n)*O(1)=O(n)

public class Only14
{
   public boolean only14(int[] nums) {
   boolean siono = true;                          O(1)
   for(int i = 0; i < nums.length; i++){          O(n)
     if(nums[i] != 1 && nums[i] != 4 ){           O(1)
       siono = false;                             O(1)
     }
   }
   return siono;                                  O(1)
   }
}
```

*O(1)+O(n)*O(1)=O(n)*

```
public class No14
{
   public boolean no14(int[] nums) {
    boolean siono = false;                        O(1)
    boolean siono2 = false;                       O(1)
    for(int i = 0; i < nums.length; i++){         O(n)
      if(nums[i] == 1){                           O(1)
        siono = true;                             O(1)
      }
      if(nums[i] == 4){                           O(1)
        siono2 = true;                            O(1)
      }
    }
    if(siono && siono2){                          O(1)
       return false;                              O(1)
     } else return true;                          O(1)
}
}
```

*O(1)+O(n)*O(1)=O(n)*

```
public class MaxSpan
{
   public int maxSpan(int[] nums) {
  int n = 0;                                      O(1)
  for(int i = 0; i<nums.length;i++){              O(n)
```

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

```
    int j = nums.length-1;                    O(1)
     while(nums[i]!= nums[j]){                O(n)
       j--;}                                  O(1)
   if(n<j-i+1){                               O(1)
   n = j-i+1;                                 O(1)
   }
  }
  return n;                                   O(1)
}
}
```

O(1)+O(n)*O(1)+O(n)*O(1)=O(n)

```
public class MaxMirror
{
   public int maxMirror(int[] nums) {
  int longi = nums.length, cont = 0, max = 0;                O(1)
  for(int i = 0; i<longi; i++){                              O(n)
    cont=0;                                                  O(1)
    for(int j = longi-1; j > -1 && i+cont < longi; j--){     O(n)
      if(nums[i+cont] == nums[j]){                           O(1)
        cont++;                                              O(1)
      } else {                                               O(1)
        if(cont>0){                                          O(1)
          max= Math.max(cont,max);                           O(1)
          cont = 0;                                          O(1)
        }
      }
    }
    max = Math.max(cont,max);                                O(1)
  }
  return max;                                                O(1)
}
}
```

O(1)+O(n)*O(1)*O(n)+O(1)=O(n^2)

```
public class HaveThree
{
public boolean haveThree(int[] nums) {
  int cont = 0;                                  O(1)
  boolean siono = false;                         O(1)
  for(int i = 0; i<nums.length;i++){             O(n)
    if(nums[i]==3 ){                             O(1)
```

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

```
    cont++;                                          O(1)
    i++;                                             O(1)
   }
   if(cont==3 && cont<=3) siono=true;      O(1)
   else siono=false;                       O(1)
  }
  return siono;                            O(1)
}
}
```

O(1)+O(n)*O(1)=O(n)

```
public class Has77
{
   public boolean has77(int[] nums) {
  boolean siono = false;                   O(1)
  for(int i = 0; i < nums.length-1; i++){  O(n)
   if(nums[i] == 7 && nums[i+1] == 7){     O(1)
   siono = true;                           O(1)
  }
  }
  for(int i = 0; i < nums.length-2; i++){  O(n)
   if(nums[i] == 7 && nums[i+2] == 7){     O(1)
   siono = true;                           O(1)
  }
  }
  return siono;                            O(1)
}
}
```

O(1)+O(n)*O(1)*O(n)*O(1)=O(n)

```
public class Fix45
{
   public int[] fix45(int[] nums) {
  for(int i = 0; i<nums.length;i++){                                  O(n)
   if((nums[i]==5 && i==0) ||(nums[i]==5 && nums[i-1]!=4)){            O(1)
    int posi5 = i;                                                    O(1)
    for(int j = 0; j<nums.length;j++){                                O(n)
     if(nums[j]==4 && nums[j+1]!=5){                                  O(1)
      int temp = nums[j+1];                                           O(1)
      nums [j+1] = 5;                                                 O(1)
      nums[posi5] = temp;                                             O(1)
      break;
```

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

```
      }
     }
    }
   }
   return nums;                                                    O(1)
}
}
```

O(n)+O(1)*O(n)+O(1)=O(n)

```
public class Fix34
{
   public int[] fix34(int[] nums) {
  int temp;                                          O(1)
  int n = 0;                                         O(1)
  for(int i = 0; i<nums.length;i++){                 O(n)
    if(nums[i]==3){                                  O(1)
      temp = nums[i+1];                              O(1)
      for(int j = n; j<nums.length;j++){             O(n)
        if(nums[j]==4){                              O(1)
          nums[j]=temp;                              O(1)
          n = j+1;                                   O(1)
          j=nums.length;                             O(1)
        }
      }
      nums[i+1]=4;                                   O(1)
    }
  }
  return nums;                                       O(1)
}
}
```

O(1)+O(n)*O(1)*O(n)*O(1)=O(n^2)

```
public class CanBalance
{
   public boolean canBalance(int[] nums) {
  for(int i = 0; i<nums.length;i++){                 O(n)
    int sum = 0;                                     O(1)
    for(int j = 0; j < i; j++)                       O(n)
      sum+=nums[j];                                  O(1)
    for(int j = i; j<nums.length; j++)               O(n)
      sum-=nums[j];                                  O(1)
    if(sum==0)                                       O(1)
    return true;                                     O(1)
```

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

```
  }
  return false;                                    O(1)
}
}          O(n)+O(1)*O(n)*O(1)*O(n)+O(1)=O(n^3)
```

3.6 Básicamente N y/o M determina la longitud del arreglo

4) Simulacro de Parcial

4.1 *C*
4.2 *B*
4.3 *A,D*
4.4 *C*
4.5 *A (a)Si*
4.6 *T(100) = T(10000) = 100m/s*
4.7 *1 y 3*
4.8 *A*
4.9 *A*
4.10 *A*
4.11 *B*
4.12 *D*
4.13 *C*
4.14 *A*

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473