ESTRUCTURA DE DATOS 1
Código ST0245

Laboratorio Nro. X
Escribir el tema del laboratorio

Susana Alvarez Zuluaga
Universidad Eafit
Medellín, Colombia
salvarezz1@eafit.edu.co

MariaJose Franco Orozco
Universidad Eafit
Medellín, Colombia
mfrancoo@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

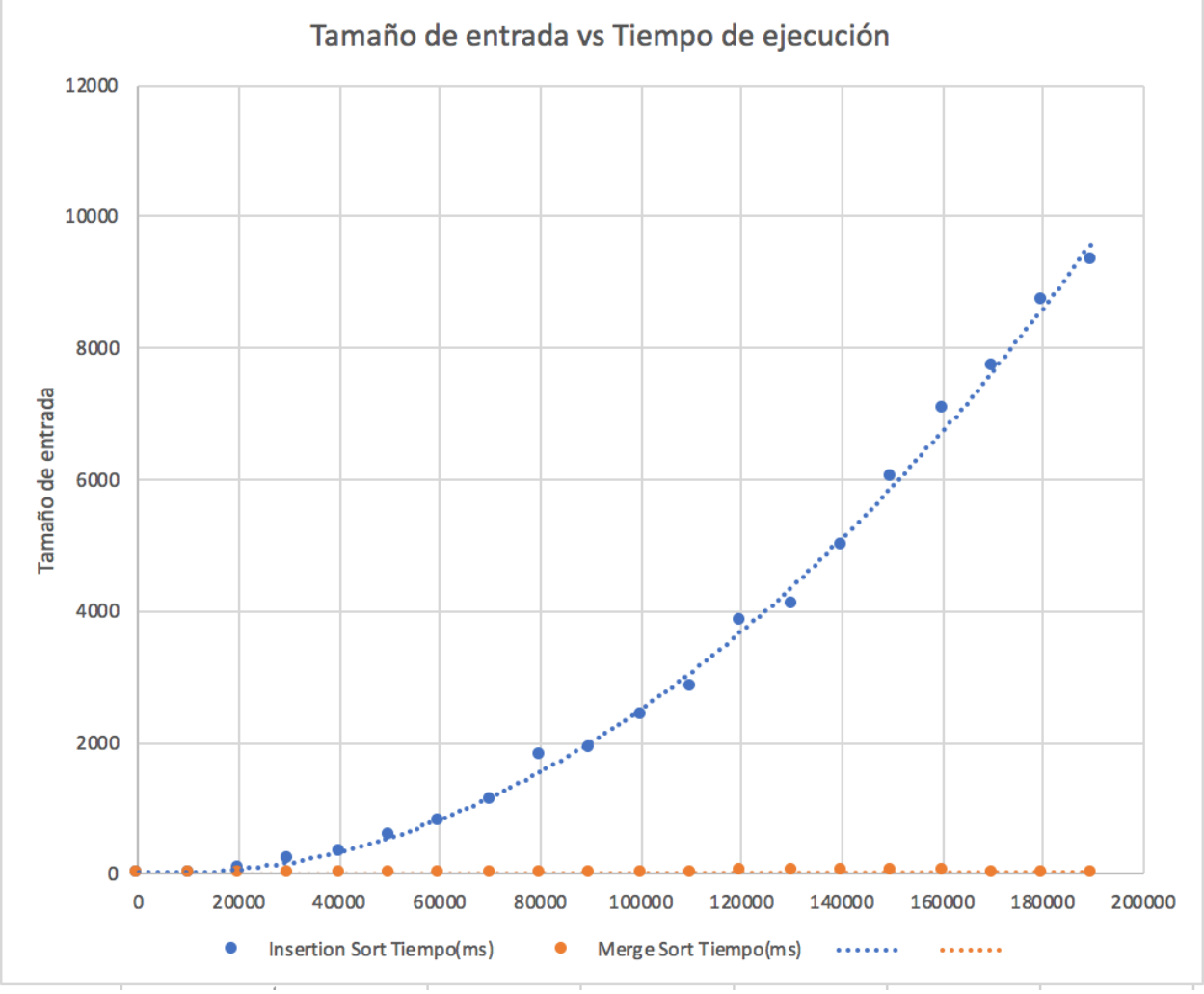
3.1

Insertion Sort	
N(numero de elementos en el arreglo)	Tiempo(ms)
0	0
10000	24
20000	105
30000	231
40000	346
50000	574
60000	799
70000	1121
80000	1806
90000	1919
100000	2434
110000	2840
120000	3842
130000	4116
140000	5009
150000	6043
160000	7067
170000	7704
180000	8730
190000	9329

Merge Sort	
N(numero de elementos en el arreglo)	Tiempo(ms)
0	0
10000	7
20000	10
30000	18
40000	19
50000	15
60000	15
70000	17
80000	21
90000	23
100000	22
110000	29
120000	41
130000	38
140000	55
150000	48
160000	53
170000	27
180000	26
190000	24

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

3.2



3.3. El algoritmo merge sort es muchísimo más eficiente que el de insertion sort cuando se están utilizando arreglos muy grandes. En la gráfica y en las tablas se ve que el tiempo en el algoritmo merge sort crece de forma más "lineal" que el tiempo en el algoritmo de insertion sort que crece polinomialmente.

3.4. El método de ordenamiento insertion sort no es lo suficientemente eficiente para ser utilizado en video juegos con millones de elementos ya que se tardaría demasiado y como vemos en la gráfica a medida que el tamaño de entrada incrementa, el tiempo crece cada vez más rápido.

3.5 Cuando se están ordenando arreglos muy grandes, en ningún caso insertion sort es más rápido que merge sort. Para que insertion sort sea más rápido que merge sort, se necesita que los arreglos tengan muy poquitos elementos y no estén muy desordenados.

3.6 El algoritmo de MaxSpan consta de 2 ciclos anidados. El ciclo de afuera lo que hace es recorrer el arreglo desde la primera posición y el ciclo de adentro empieza a recorrer el arreglo desde la ultima posicion. El algoritmo entra al primer ciclo y coge un elemento del arreglo, luego entra al ciclo anidado y empieza a buscar desde la ultima posición del arreglo si hay un

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

elemento igual al que se cogió en el ciclo de afuera. Si se encuentra uno, se guarda el "span" en una variable. El "span" esta dado por la cantidad de elementos que hay entre los dos valores que son iguales incluyéndolos a ellos mismos. Al guardar el span, se hace un break porque solo nos interesa el span más grande que hay entre 2 elementos. Ahí estaríamos encontrando el span entre 2 elementos que son iguales, pero para encontrar el maxSpan que sería la distancia máxima que hay entre cualquiera 2 elementos iguales en TODO el arreglo, el algoritmo al salirse del ciclo que esta anidado, pregunta si el span es mas grande que el maxSpan, y si si lo es, el maxSpan toma el valor del span que se acaba de encontrar.

3.7

2.1) Array 2

1. Shift left-O(n)
2. notAlone-O(n)
3. has77-O(n)
4. countEvens-O(n)
5. Sum13-O(n)

2.2) Array 3

1.linearIn-O(n^2)
2. countClumps- O(n^2)
3. MaxSpan- O(n^2)
4. seriresUp- O(n^2)
5. fix34- O(n^2)

3.8 La variable n es el numero de veces que se van a realizar cierta operación; por ejemplo, cuando hay un ciclo, n va a representar la condición de parada del ciclo, la cual la mayoría de las veces corresponde a la longitud del arreglo, lo que significa que el ciclo se va a repetir las veces que sea necesario hasta que llegue a la condición de parada (n); por esto decimos que el ciclo se repite n veces.

4) Simulacro de Parcial

4.1* c) *O(n+m)
4.2 **d)O(m×n)**
4.3 **b) O(ancho)**
4.4 b) O(n^3)
4.5 d) O(n^2)
4.6 a) T(n)=T(n−1)+C
4.9 d) Ejecuta más de $n \times m$ pasos.
4.10 d) Ejecuta exactamente pasos
4.11 c) Ejecuta $T(n)=T(n-1)+T(n-2)+c$ pasos.
4.12 a)Sí
4.13 c)T(n) = 2T(n/2) + n
4.14 b) Ejecuta $O(n.m)$ instrucciones

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