# **Laboratory practice No. 2:**

# **Algorithm Complexity.**

|  |  |
| --- | --- |
| **Jose Manuel Fonseca Palacio**  Universidad Eafit  Medellín, Colombia  jmfonsecap@eafit.edu.co | **Santiago Puerta Florez**  Universidad Eafit  Medellín, Colombia  spuertaf@eafit.edu.co |

***1.1)***

***2.1) and 2.2)*** Bowth of them are solved in the github page.

***3.1)***

Time table for inserionSort()

|  |  |
| --- | --- |
| Array Length | Time (Milliseconds) |
| 100000 | 4604 |
| 101500 | 5125 |
| 103000 | 5070 |
| 104500 | 5095 |
| 106000 | 5117 |
| 107500 | 5215 |
| 109000 | 5822 |
| 110500 | 6977 |
| 112000 | 6813 |
| 113500 | 5902 |
| 115000 | 6510 |
| 116500 | 6235 |
| 118000 | 6277 |
| 119500 | 7691 |
| 121000 | 8790 |
| 122500 | 9319 |
| 124000 | 8199 |
| 125500 | 7459 |
| 127000 | 7427 |
| 128500 | 8167 |

Time table for mergeSort()

|  |  |
| --- | --- |
| Array Length | Time (Milliseconds) |
| 3551000 | 6926 |
| 3576500 | 8437 |
| 3602000 | 8116 |
| 3627500 | 8696 |
| 3653000 | 6194 |
| 3678500 | 9706 |
| 3704000 | 6001 |
| 3729500 | 6114 |
| 3755000 | 5372 |
| 3780500 | 8508 |
| 3806000 | 5772 |
| 3831500 | 4606 |
| 3857000 | 2351 |
| 3882500 | 2227 |
| 3908000 | 2296 |
| 3933500 | 2378 |
| 3959000 | 2433 |
| 3984500 | 2350 |
| 4010000 | 2433 |
| 4035500 | 2475 |

***3.2)***

Graph for insertionSort()

Graph for mergeSort()

***3.3)*** No, it wouldn’t be appropiate to use insertion sort for such a hard job involving tons of data, due to it’s complexity (O ()). For an-array of 50 million deorganized elements the algorithm would take aproximately 1 hour to sort all elements.

***3.4)***

***3.5)*** For big arrays if you are wishing insertionSort to be faster than mergeSort the data given to the method insertionSort must be all the same. With all numbers equal to each other insertionSort give us the following table.

|  |  |  |
| --- | --- | --- |
| insertionSort | mergeSort |  |
| Time (Milliseconds) | Time (Milliseconds) | Array Length |
| 94 | 1513 | 4500000 |
| 24 | 2040 | 6000000 |
| 12 | 3760 | 7500000 |
| 40 | 7060 | 9000000 |

Curiosly if you give the mergeSort an already sorted array, the method takes 0 milliseconds to give a response, in contrary with the insertionSort method that takes more time to give an answer than the time taken by the mergeSort. Giving us the following table.

|  |  |  |
| --- | --- | --- |
| insertionSort | mergeSort |  |
| Time | Time | Array Length |
| 76 | 0 | 4500000 |
| 14 | 0 | 6000000 |
| 14 | 0 | 7500000 |
| 11 | 0 | 9000000 |

***4) Practice for midterms***

* 1. *a*
  2. *b*
  3. *length-1*

***5) Recommended reading (optional)***

Mapa conceptual

**6)** **Team work and gradual progress (optional)**

* 1. Meeting minutes

***6.2*** History of changes of the code

***6.3*** History of changes of the report