Algorithm

< Comparison of Quicksort and Heapsort>

Signature:

Signature:

Signature:

2017011976 박경리  
2017012033 배수연  
2017012197 여채린

Participant No.1: 2017011976 박경리

Participant No.2: 2017012033 배수연

Participant No.3: 2017012197 여채린

2017011976 박경리  
2017012033 배수연  
2017012197 여채린

**1. Implementing the two sorting algorithms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n** | **Comparison** | | **Exchange** | |
| **Quick** | **Heap** | **Quick** | **Heap** |
| **100** | 784 | 677 | 193 | 585 |
| **200** | 1832 | 1643 | 441 | 1361 |
| **500** | 5095 | 5131 | 1246 | 4061 |
| **1000** | 12252 | 11685 | 2693 | 9108 |
| **2000** | 26260 | 26351 | 5816 | 20156 |
| **3000** | 43340 | 42076 | 9017 | 32063 |
| **4000** | 60551 | 58672 | 12312 | 44300 |
| **5000** | 78885 | 75609 | 15651 | 57082 |

In number of comparisons, the two sorting algorithms showed almost similar efficiencies. But heapsort is very inefficient compared to quicksort in number of exchanges.

**2. Show the wall-clock execution times as well.**

<QuickSort>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **1st Try** | **2nd Try** | **3rd Try** | **4th Try** | **5th Try** | **Average** |
| **100** | 0.014 | 0.017 | 0.015 | 0.014 | 0.014 | 0.015 |
| **200** | 0.028 | 0.027 | 0.028 | 0.028 | 0.028 | 0.028 |
| **500** | 0.069 | 0.071 | 0.069 | 0.070 | 0.069 | 0.070 |
| **1000** | 0.145 | 0.145 | 0.141 | 0.146 | 0.148 | 0.145 |
| **2000** | 0.305 | 0.303 | 0.302 | 0.308 | 0.303 | 0.304 |
| **3000** | 0.481 | 0.476 | 0.468 | 0.480 | 0.474 | 0.476 |
| **4000** | 0.648 | 0.656 | 0.650 | 0.649 | 0.642 | 0.649 |
| **5000** | 0.816 | 0.834 | 0.833 | 0.823 | 0.824 | 0.826 |

//in milli seconds

<HeapSort >

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **1st Try** | **2nd Try** | **3rd Try** | **4th Try** | **5th Try** | **Average** |
| **100** | 0.024 | 0.023 | 0.025 | 0.024 | 0.022 | 0.024 |
| **200** | 0.043 | 0.044 | 0044 | 0.047 | 0.044 | 0.044 |
| **500** | 0119 | 0.120 | 0.120 | 0.120 | 0.121 | 0.120 |
| **1000** | 0.0256 | 0.215 | 0.216 | 0.213 | 0.218 | 0.224 |
| **2000** | 0.462 | 0.464 | 0.464 | 0.461 | 0.464 | 0.463 |
| **3000** | 0.809 | 0.724 | 0.728 | 0.722 | 0.726 | 0.742 |
| **4000** | 0.870 | 0.997 | 0.995 | 1.106 | 00998 | 0.993 |
| **5000** | 1.273 | 1.272 | 1.271 | 1.276 | 1.289 | 1.276 |

//in seconds

Both algorithms are close to the theoretical values.

Quicksort is much faster than heapsort.

**#Used Code**

<Quicksort>



<Heapsort>

