**Group Members:**

**Nishant Shekhar (nxs167130)**

**Anusha Agasthi (nxa162430)**

**Prince Patel (pap160930)**

**Question 1.**

**Comparison of Quick Sort performance using first partition. NOT Hoare algorithm**

|  |  |  |
| --- | --- | --- |
| No. of elements | Running Time/Memory  With random , distinct elements | Running Time/Memory  With array in descending order. |
| 8M | Time: 3486 msec.  Memory: 340 MB / 981 MB. | Time: 3112 msec.  Memory: 556 MB / 1237 MB. |
| 16M | Time: 8701 msec.  Memory: 682 MB / 1237 MB. | Time: 7556 msec.  Memory: 671 MB / 1749 MB. |
| 32M | Time: 17271 msec.  Memory: 835MB / 1900MB. | Time: 15954 msec.  Memory: 1032 MB / 2604 MB. |
| 52M | Time: 53093 msec.  Memory: 1602 MB / 2813 MB. | Time: 61917 msec.  Memory: 1272 MB / 3093 MB. |

**Comparison of Quick Sort performance using first partition. Hoare algorithm**

|  |  |  |
| --- | --- | --- |
| No. of elements | Running Time/Memory  With random , distinct elements | Running Time/Memory  With array in descending order. |
| 8M | Time: 3783 msec.  Memory: 340 MB / 981 MB. | Time: 3024 msec.  Memory: 556 MB / 1237 MB. |
| 16M | Time: 8293 msec.  Memory: 682 MB / 1237 MB. | Time: 7645 msec.  Memory: 671 MB / 1749 MB. |
| 32M | Time: 17467 msec.  Memory: 844 MB / 19003 MB. | Time: 15773 msec.  Memory: 1082MB / 2627 MB. |
| 52M | Time: 52259 msec.  Memory: 1601 MB / 2799 MB. | Time: 62457 msec.  Memory: MB / MB. |

**Question 2:**

**Comparison of Quick Sort performance using dual pivot partition.**

|  |  |  |
| --- | --- | --- |
| No. of elements | Running Time/Memory  With random , distinct elements | Running Time/Memory  With array in descending order. |
| 8M | Time: 3144 msec.  Memory: 158 MB / 981 MB. | Time: 2726 msec.  Memory: 178 MB / 1237 MB. |
| 16M | Time: 5853 msec.  Memory: 306 MB / 981 MB. | Time: 5811 msec.  Memory: 461 MB / 1237 MB. |
| 32M | Time: 14598 msec.  Memory: 618 MB / 1709 MB. | Time: 15954 msec.  Memory: 1032 MB / 2604 MB. |
| 52M | Time: 22702 msec.  Memory: 994 MB / 1702 MB. | Time: 25879 msec.  Memory: 1766 MB /2819 MB. |

**Question 3:**

Here as the number of K increased the time taken by Select algorithm decreased than max and min heap. Then min heap performed a bit better than the Max heap.

**Question 4:**

**Comparison of best version of my merge sort with best version of my quick sort.**

|  |  |  |
| --- | --- | --- |
| No. of elements | Running Time/Memory  With random, distinct elements  For Quick Sort | Running Time/Memory  With random , distinct elements for Merge Sort |
| 100 | Time: 0 msec.  Memory: 2 MB / 245 MB. | Time: 16 msec.  Memory: 2 MB / 245 MB. |
| 1K | Time: 0 msec.  Memory: 2 MB / 245 MB. | Time: 95 msec.  Memory: 2 MB / 245 MB. |
| 10K | Time: 4 msec.  Memory: 2 MB / 245 MB. | Time: 56373 msec.  Memory: 2 MB / 245 MB. |

We can merge sort started becoming very slow from the third result . Most probably because of making copy big array to another.

|  |  |  |
| --- | --- | --- |
| No. of elements | Running Time/Memory  With random, distinct elements  For Quick Sort | Running Time/Memory  With random , distinct elements for Merge Sort |
| 100 | Time: 0 msec.  Memory: 2 MB / 245 MB. | Time: 6 msec.  Memory: 2 MB / 245 MB. |
| 1K | Time: 0 msec.  Memory: 2 MB / 245 MB. | Time: 96 msec.  Memory: 2 MB / 245 MB. |
| 10K | Time: 0 msec.  Memory: 2 MB / 245 MB. | Time: 58835 msec.  Memory: 2 MB / 245 MB. |

We see the same result , Merge Sort starts slowing down quickly with the increase of input size