**Report (SP5)**

**Group : g39**

**Authors: Khushbu Patil, Vatsal Patel, Shruti Shetye**

**Q1. Comparison of performance of the two versions of partition discussed in class on the running time of Quick sort.**

|  |  |  |  |
| --- | --- | --- | --- |
| Array Type | n | Time to quicksort by Partition 1 | Time to quicksort by Partition 2 |
| Random | 10M | 360 msec. | 250 msec. |
| Descending | 10M | 187 msec. | 187 msec. |
| Random | 15M | 437 msec. | 359 msec. |
| Descending | 15M | 266 msec. | 266 msec. |
| Random | 20M | 5183 msec. | 5068 msec. |
| Descending | 20M | 3654 msec. | 3937 msec. |
| Random | 50M | 13504 msec. | 13668 msec. |
| Descending | 50M | 9274 msec. | 9882 msec. |
| Random | 70M | 18735 msec. | 19590 msec. |
| Descending | 70M | 13494 msec. | 13475 msec. |
| Random | 100M | 28530 msec. | 27733 msec. |
| Descending | 100M | 18739 msec. | 20594 msec. |

Observation:

Algorithm 2 gives some performance improvement for arrays with randomly sorted numbers of smaller sizes(upto 25 M) but comparable for larger sizes.

However, when the array is sorted in descending order, it performs comparably with algorithm 1 for small sizes but poorer for larger sizes.

**Q3. Comparison of performance of the three versions of select algorithm discussed in class**

|  |  |  |  |
| --- | --- | --- | --- |
| n | Priority queue(max-heap) | Priority queue(min-heap) | O(n) select algorithm |
| 10000000 | 7154 | 100 | 121 |
| 20000000 | 19624 | 152 | 129 |
| 25000000 | 20955 | 174 | 173 |
| 30000000 | 22110 | 219 | 748 |
| 50000000 | 36302 | 338 | 1845 |

As we can see, versions-2(min-heap) and 3(O(n) algorithm) gives significant improvements over version-1 (max-heap). Version-3 is supposed to be working best with respect to running time but our experiments show that it doesn’t. That maybe because of the randomized behavior of the algorithm.