BLG 335E

Analysis

of

Algoritms 1

Fall 2015

Homework 3

Ozan Özyeğen

150120115

1. The execution times of each sorting algorithm is tested.

|  |  |  |  |
| --- | --- | --- | --- |
| Sorting Type | QuickSort | CountingSort | RadixSort |
| 1k-10k | 0.000426 | 0.000335 | 0.000507 |
| 1k-10M | 0.000441 | 0.234741 | 0.000793 |
| 100k-10k | 0.039356 | 0.006369 | 0.032041 |
| 100k-10M | 0.042134 | 0.132663 | 0.046738 |

* 1. If our algorithm is randomized, the way we select our pivot is not going to affect the computational complexity. However, if we select the last element as a pivot and the array is reverse-ordered than the computational complexity is going to be O(n^2) instead of O(n.logn).
  2. Reversed order array is the worst case of implementation for the quicksort. The time complexity of Quicksort is O(n^2).

1. In Counting Sort time complexity O(n+k), k stands for the biggest value in the array. We have to create a temporary array which has the size of the biggest value. Therefore k affects the time complexity of the Counting Sort.
2. The time complexity of Radix Sort for all cases is O(d.n). “d” stands for the number of digits that maximum number has. So if the array has the maximum number of 10 billion instead of 10 thousand, it greatly effects the calculation time of the sorting algorithm.