Prashant Kumar

Quentin Jimenez

Noah Malhi

References:

|  |
| --- |
|  |
|  | https://www.geeksforgeeks.org/shuffle-a-given-array-using-fisher-yates-shuffle-algorithm/ |
|  | https://www.cs.upc.edu/~conrado/research/reports/ALCOMFT-TR-03-50.pdf |
|  | Partial QuickSort- Conrado Martinez |

METHODS

1. Fischer Yates shuffling was used to shuffle the array using uniform random int generator.

2. Randomized versions of Lomuto and Hoare partitioning schemes were created and tested for an array for size up to 10^8 elements.

3. partialQuicksort method was used with arbitrary k to sort elements such that each unsorted element was at most k elements far from the pivot.

4. Knuth Quicksort employed partialQuicksort and insertionSort consecutively to sort the almost sorted array.

Sys config:

Windows 10 pro 64bit

AMD Ryzen 7 3700x 8-Core Processor 3693Mhz 8 cores 16 logical processors

n = 10^7

tolerance = 5

Optimum k is taken from the average of k1 and k2, k1 is initialized as 10 and k2 as 1000.

#Elements in array: 10000000

Optimum K: 505

Runtime in us: 1.00128e+06 for k= 389

Runtime in us: 1.16404e+06 for k= 621

Optimum K: 315

Runtime in us: 904182 for k= 244

Runtime in us: 1.00444e+06 for k= 389

Optimum K: 199

Runtime in us: 866513 for k= 155

Runtime in us: 895937 for k= 244

Optimum K: 127

Runtime in us: 830480 for k= 100

Runtime in us: 889865 for k= 155

Optimum K: 82

Runtime in us: 832453 for k= 66

Runtime in us: 835155 for k= 100

Optimum K: 55

Runtime in us: 847283 for k= 45

Runtime in us: 834445 for k= 66

Optimum K: 72

Runtime in us: 838952 for k= 66

Runtime in us: 834680 for k= 78

Optimum K: 83

Runtime in us: 843192 for k= 78

Runtime in us: 834970 for k= 87

Optimum K: 89

Runtime in us: 831949 for k= 87

Runtime in us: 833644 for k= 91

Optimum K: 84

Runtime in us: 844360 for k= 83

Runtime in us: 833219 for k= 87

Optimum K: 87

Runtime in us: 866136 for k= 87

Runtime in us: 834232 for k= 87

Optimum K: 89

Optimum k is : 89