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 = 3

Optimum k is taken from the average of k1 and k2, k1 is initialized as 1 and k2 as 1000.

$ ./main.exe

#Elements in array: 10000000

Shuffled Array:

Runtime in microseconds: 771938 for k = 383

Runtime in microseconds: 898075 for k = 618

Optimum K: 309

Runtime in microseconds: 691285 for k = 237

Runtime in microseconds: 763625 for k = 383

Optimum K: 192

Runtime in microseconds: 653435 for k = 147

Runtime in microseconds: 690116 for k = 237

Optimum K: 119

Runtime in microseconds: 646815 for k = 92

Runtime in microseconds: 642923 for k = 147

Optimum K: 164

Runtime in microseconds: 655719 for k = 147

Runtime in microseconds: 675507 for k = 181

Optimum K: 136

Runtime in microseconds: 649758 for k = 126

Runtime in microseconds: 646542 for k = 147

Optimum K: 153

Runtime in microseconds: 660522 for k = 147

Runtime in microseconds: 646459 for k = 159

Optimum K: 164

Runtime in microseconds: 649768 for k = 159

Runtime in microseconds: 658786 for k = 168

Optimum K: 157

Runtime in microseconds: 642563 for k = 156

Runtime in microseconds: 661991 for k = 159

Optimum K: 153

Runtime in microseconds: 662739 for k = 152

Runtime in microseconds: 652822 for k = 156

Optimum K: 155

Runtime in microseconds: 652681 for k = 156

Runtime in microseconds: 646460 for k = 156

Optimum K: 157

Optimum k is : 157

Shuffled Array :

Runtime in microseconds: 660998 for k = 157

Sorted Array :

0 1 2 3 4 ...9999995 9999996 9999997 9999998 9999999