

Tukey's ninther

Tukey proposed an idea (see quick sort lecture slides) to find a better median in random arrays so that quick sort uses less comparisons. It gets approximate median of 9 items from the data array to find a pivot item. This assignment is to test if a quick sort initiated with "Tukey's ninther's " is faster than regular quick sort algorithm. You need to compare these two approaches in the sorting of 256K 4-char strings, randomly created between aaaa - ZZZZ. Random strings have only a-z, A-Z letters. Repeat the comparison 100 times and report

- Average running time in milliseconds of Tukey's approach and regular Quick Sort.
- How many times Tukey's approach is faster than regular Quick Sort.
- What is t-value? Assuming run times are normally distributed with same variance, use t-test to compare their normal distributions. Please see https://en.wikipedia.org/wiki/Student%27s_t-test#Independent_two-sample_t-test

Your program should report above metrics as output, similar to

```
Tukey QS Ave: 125 ms, Reg. QS: 129 ms  
Tukey QS was faster in 51% of 100 runs.  
T-value for two distributions is 2.26.
```

Hint: Do not forget to randomize the data; and make sure that both algorithms sort same data for comparison.

You can see how Tukey's ninther is used in a java implementation [here](#).

Do not include
memcpy,
#include <bits/stdc++.h>.

Do not include unnecessary headers and functions. Code syntax will be graded.

Submission Procedure:

Submit a single complete (compiles and runs) .cpp file with the following format:
"Lastname_FirstInitial_LastFourDigitsOfCWID_L6.cpp", such as Smith_J_5678_L7.cpp.