

Zhenyu Yan

8/27/2018

HW1 report

The contrast on performance of my program on nike.cs.uga.edu program that provide:

5 number

```
[zhenyu vcf5-HW1] time ./statistics < test/05numbers.txt
range[-100 5]
median index(floor) = 3
the mean = -42.600000 (-213/5)
the median = -100 at index 2 (odd)
the mode = -100 (2 occurrences)

real    0m0.010s
user    0m0.000s
sys     0m0.001s
[zhenyu vcf5-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/05numbers.txt
range [-100 5]
median index (floor) = 3
the mean = -42.600000 (-213/5)
the median = -22 at index 2 (odd)
the mode = -100 (2 occurrences)

real    0m0.010s
user    0m0.000s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is no difference.

6 number

```
[zhenyu nike-HW1] time ./statistics <test/6numbers.txt
range[1 6]
median index(floor) = 3
the mean = 3.666667 (22/6)
the median = 4.000000 at index 3 and index 4 (even)
the mode = 4 (2 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/6numbers.txt
range [1 6]
median index (floor) = 3
the mean = 3.666667 (22/6)
the median = 4 at index 3 and index 4 (even and floored)
the mode = 4 (2 occurrences)

real    0m0.002s
user    0m0.000s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is the same.

10 number

```
[zhenyu nike-HW1] time ./statistics <test/10numbers.txt
range[24 85]
median index(floor) = 5
the mean = 60.800000 (608/10)
the median = 64.500000 at index 5 and index 6 (even)
the mode = 46 (2 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/10numbers.txt
range [24 85]
median index (floor) = 5
the mean = 60.800000 (608/10)
the median = 73 at index 5 and index 6 (even and floored)
the mode = 46 (2 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
[zhenyu nike-HW1]
```

We can see here the difference between my program and the program that provide is no difference. But in here, the median is different from my program, I think the reason is our algorithm is different, and I saw many student have the different result in the group me too.

10-multi mode

```
[zhenyu nike-HW1] time ./statistics <test/10-multimode-numbers.txt
range[3 97]
median index(floor) = 20
the mean = 44.384615 (1731/39)
the median = 33.000000 at index 19 (odd)
the mode = 54 (6 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/10-multimode-numbers.txt
range [3 97]
median index (floor) = 20
the mean = 44.384615 (1731/39)
the median = 38 at index 19 (odd)
the mode = 54 (6 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is no difference.

29 number

```
[zhenyu vcf5-HW1] time ./statistics < test/29numbers.txt
range[3 97]
median index(floor) = 15
the mean = 39.965517 (1159/29)
the median = 32 at index 14 (odd)
the mode = 28 (3 occurrences)

real    0m0.008s
user    0m0.001s
sys     0m0.000s
[zhenyu vcf5-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/29numbers.txt
range [3 97]
median index (floor) = 15
the mean = 39.965517 (1159/29)
the median = 32 at index 14 (odd)
the mode = 28 (3 occurrences)

real    0m0.009s
user    0m0.001s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is very little.(mine is 0.01 second faster).

40 number

```
[zhenyu nike-HW1] time ./statistics <test/40numbers.txt
range[-5 20]
median index(floor) = 20
the mean = 7.775000 (311/40)
the median = 9.500000 at index 20 and index 21 (even)
the mode = 10 (4 occurrences)

real    0m0.002s
user    0m0.000s
sys     0m0.001s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/40numbers.txt
range [-5 20]
median index (floor) = 20
the mean = 7.775000 (311/40)
the median = 9 at index 20 and index 21 (even and floored)
the mode = 10 (4 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is the same.

40-sort

```

[zhenyu nika-HW1] time ./statistics <test/40-sorted.txt
range[-5 20]
median index(floor) = 20
the mean = 7.775000 (311/40)
the median = 9.500000 at index 20 and index 21 (even)
the mode = 10 (4 occurrences)

real    0m0.002s
user    0m0.001s
sys      0m0.001s
[zhenyu nika-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/40-sorted.txt
range [-5 20]
median index (floor) = 20
the mean = 7.775000 (311/40)
the median = 9 at index 20 and index 21 (even and floored)
the mode = 10 (4 occurrences)

real    0m0.002s
user    0m0.000s
sys      0m0.001s

```

We can see here the difference between my program and the program that provide is no difference. Because this is an already sort array, if we use quick sort, the time complexity will become $O(n^2)$. If we can check if the input number is sorted or not, we can make it faster.

41 number

```

[zhenyu vcf5-HW1] time ./statistics < test/41numbers.txt
range[-5 20]
median index(floor) = 21
the mean = 8.024390 (329/41)
the median = 9 at index 20 (odd)
the mode = 10 (4 occurrences)

real    0m0.009s
user    0m0.000s
sys      0m0.001s
[zhenyu vcf5-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/41numbers.txt
range [-5 20]
median index (floor) = 21
the mean = 8.024390 (329/41)
the median = 10 at index 20 (odd)
the mode = 10 (4 occurrences)

real    0m0.010s
user    0m0.001s
sys      0m0.000s

```

We can see here the difference between my program and the program that provide is very little.(mine is 0.01 second faster).

10000number

```

[zhenyu nika-HW1] time ./statistics <test/10000numbers.txt
range[-800 1000]
median index(floor) = 4997
the mean = 99.170319 (991009/9993)
the median = 101.000000 at index 4996 (odd)
the mode = -249 (17 occurrences)

real    0m0.008s
user    0m0.007s
sys      0m0.001s
[zhenyu nika-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/10000numbers.txt
range [-800 1000]
median index (floor) = 4997
the mean = 99.170319 (991009/9993)
the median = 101 at index 4996 (odd)
the mode = -249 (17 occurrences)

real    0m0.006s
user    0m0.006s
sys      0m0.000s

```

We can see here the difference between my program and the program that provide is very little.(mine is 0.02 second slower). But we can see as the number of value become larger, the time for my program become slower.

100000number

```
[zhenyu nike-HW1] time ./statistics <test/100000numbers.txt
range[-800 1000]
median index(floor) = 49969
the mean = 102.989473 (10292459/99937)
the median = 104.000000 at index 49968 (odd)
the mode = 644 (82 occurrences)

real    0m0.062s
user    0m0.059s
sys     0m0.002s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/100000numbers.txt
range [-800 1000]
median index (floor) = 49969
the mean = 102.989473 (10292459/99937)
the median = 104 at index 49968 (odd)
the mode = 644 (82 occurrences)

real    0m0.042s
user    0m0.040s
sys     0m0.002s
```

For 100000 numbers, my programs runs 0.20 second slower. My algorithm is slower because the way I know the range of the input array is to create a file and writes all the previous input. (in the meantime, I count how many values) Then read it again. I think if I can use arraylist, linked list, the program will become faster.

Med

```
[zhenyu vcf5-HW1] time ./statistics < test/med.txt
range[-5 20]
median index(floor) = 21
the mean = 8.073171 (331/41)
the median = 9 at index 20 (odd)
the mode = 10 (4 occurrences)

real    0m0.008s
user    0m0.000s
sys     0m0.001s
[zhenyu vcf5-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/med.txt
range [-5 20]
median index (floor) = 21
the mean = 8.073171 (331/41)
the median = 10 at index 20 (odd)
the mode = 10 (4 occurrences)

real    0m0.012s
user    0m0.000s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is very little.(mine is 0.04 second faster).

Op

```
[zhenyu nike-HW1] time ./statistics <test/op.txt
range[5 999999]
median index(floor) = 2
the mean = 250022.250000 (1000089/4)
the median = 42.500000 at index 2 and index 3 (even)
the mode = 5 (1 occurrences)

real    0m0.002s
user    0m0.000s
sys     0m0.002s
[zhenyu nike-HW1] time ~maria/public_html/classes/x730-Spring-2018/1-assignment/statistics < test/op.txt
range [5 999999]
median index (floor) = 2
the mean = 250022.250000 (1000089/4)
the median = 78 at index 2 and index 3 (even and floored)
the mode = 5 (1 occurrences)

real    0m0.002s
user    0m0.001s
sys     0m0.001s
```

We can see here the difference between my program and the program that provide is very little.(mine is 0.02 second slower).

Overall, C is a bit different from C++ and java. I need to study and learn more about it. That is the reason I do not code this program well and use some very slow algorithm to finish. There are still a lot of improve that I mentions above to reduce the time complexity and space complexity.