Description of test cases used

k=n/2; entries in the table is time; n is the power of 2

1)number been processed is range from 0 to 100

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
| quicksort | 0(s) |  |  |  |  |  |  |  |  |  |
| quickselection |  |  |  |  |  |  |  |  |  |  |
| Group of 5 | 0 |  |  |  |  |  |  |  |  | 0.000105 |
| Group of 7 | 0 | 0 | 0 |  |  |  |  |  |  | 0.000104 |
| Group of 3 | 0 | 0 |  | 0 | 0 |  |  |  |  | 0.000118 |
|  |  |  |  |  |  |  |  |  |  |  |
| quicksort | 0.00016 | 0.00034 | 0.00076 | 0.001528 | 0.002851 | 0.006019 | 0.0128 | 0.025177 | 0.053725 |  |
| quickselection |  |  | 0.000124 | 0.000459 | 0.001003 | 0.001205 | 0.003782 | 0.007549 | 0.059511 |  |
| Group of 5 | 0.00016 | 0.000472 | 0.001028 | 0.001844 | 0.004421 | 0.03623 | 0.030824 | 0.067144 | 2.38368 |  |
| Group of 7 | 0.000204 | 0.000376 | 0.000932 | 0.001823 | 0.003997 | 0.008034 | 0.024913 | 0.09504 | 0.660491 |  |
| Group of 3 | 0.000253 | 0.000466 | 0.001059 | 0.003197 | 0.005449 | 0.012586 | 0.216718 | 0.694411 | 30.6393 |  |

2) number been processed is range from 0 to n

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| quicksort | 0.00944 | 0.01944 | 0.041447 | 0.084115 | 0.17405 |
| quickselection | 0.001688 | 0.003036 | 0.007268 | 0.011158 | 0.022685 |
| Group of 5 | 0.007167 | 0.013623 | 0.028125 | 0.053778 | 0.108706 |
| Group of 7 | 0.00591 | 0.013224 | 0.027624 | 0.052484 | 0.106295 |
| Group of 3 | 0.010558 | 0.021139 | 0.025801 | 0.088498 | 0.170455 |

In the above test case, data is generated by the random generator provided by the project description and is range from 1-100. And we found when there has a lot of data have same value the performance of dselect is way worse than quick select or even quick sort.

But when there are less same value, the performance for Dselect is better.

Comparative analysis and conclusions

From the results of complexity, we can see that as the group size increases, the upper bound time is decreasing. And we can prove it by the results of the program.