Project 1

Decheng Zhu –

Ruoshi Xia - 20167297

Ryn Ou – 20183436

Jack Taylor – 20100745

Experiment 1

The graphs below showed the difference between the time used for Binary Search and Trinary Search after running the code for experiment 1 for several times.

图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成

The red lines above are the time taken for Binary Search.

The blue lines above are and the time taken for Trinary Search.

For Binary Search, the recurrence of the algorithm is T(n) = T(n/2), while for Triple Search, the recurrence of the algorithm is T(n) = T(n/3).

We can obtain the time complexity of the algorithm; however, the complexity of n only represents the execution of each line, and we cannot see it in time taken. Moreover, since the speed of each computer is different, which will result to a reduction in running time, so we cannot observe this in time alone, but if for the search we add a measure of then by using this value and the input size of the algorithm we can observe that the algorithm is O(log n).

See Experiment 2 in the next page.

Experiment 2

The graphs below showed the difference between the time used for Binary Search and Trinary Search after running the code for experiment 2 for several times.

图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成

图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成

The red lines above are the time taken for Binary Search.

The blue lines above are and the time taken for Trinary Search.

Similar to Experiment 1, we can observe that the algorithm is O(log n).

See Question 1 and 2 in the next page.

Question 1

According to the experiment results above, it is hard to tell if the growth in execution time is consistent with O(log n). For the binary search results in both experiments there are graphs that appear to show O(log n) growth, however this is less clear for trinary search. We would likely need a much larger n to see this trend.

Question 2

According to the experiment results above, we can see that the difference between the

times of Binary Search and Trinary Search are not always kept within 10% of each other in both cases.

In the case of search, the time speed of Binary Search and Trinary Search depends on many factors, the time complexity of the algorithm is just a mathematical analysis, the implementation of the algorithm and the efficiency of the language also determine the time of the algorithm.

In most cases, Binary Search is slower than Trinary Search, and with the increase of n, Trinary Search will be at least 10% faster than Binary Search by n=16000. There are no conditions in which Binary Search will be at least 10% faster than Trinary Search.

“I confirm that this submission is our own work and is consistent with the Queen's regulations on Academic Integrity.”