COSC241 Assignment 1 Report **Group 6: Callum Tanner and James Cross**

Altering the size of the ExamPile

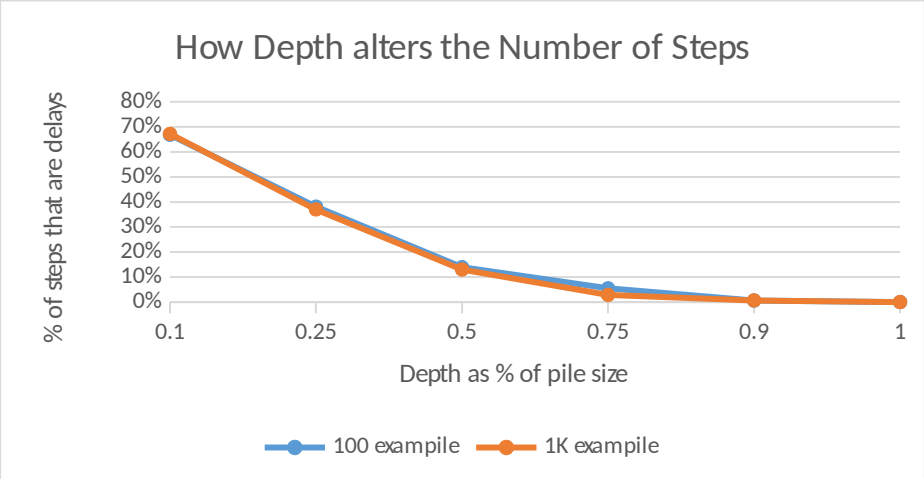
ExamPile Size	Depth	Delays	Mark	Total
10	10	0	10	10
100	10	202	100	302
1000	10	24600	1000	25600

We see from the above that the average size of the string from sortingSteps() increases as the size of the ExamPile increases, if the depth is kept constant. However, it does not increase in a linear fashion. Increasing the size of the pile from 10 to 100 (a factor of 10) does not increase the total number of steps by the same factor (10 to 302 is a factor of 30). Likewise, from 100 to 1000 increases the string by more than a factor of 10. This infers that the length of the string grows exponentially as the ExamPile increases in size.

Altering the size of depth

It is clear that the shortest possible number of steps is achieved by having a depth equal to the size of the ExamPile, since this results in all the steps being “mark” with no “delay”. In a real life situation this is not entirely practical. The ExamPile may be extremely large, making the depth equal to the size of the ExamPile would simply mean always searching through the pile to find the next exam.

The question of depth affecting steps is therefore a matter of how large the depth should be in comparison to the size of the ExamPile. Below are the results of running sortingSteps() on ExamPiles of either size 100 or 1000 at various depths, their depths are listed as a percentage of the pile size. Since delay steps only increase the number of steps, we have used the percentage of delays to measure efficiency.



As you can see, the optimal value for depth is always 100% of pile size. However, the number of delay steps drops to less than 10% at a depth of 75% of the pile size and is negligible once this reaches 90%.

Therefore, in a real-life scenario like the one described in the assignment outline, the optimal length should be in that 75-90% range.