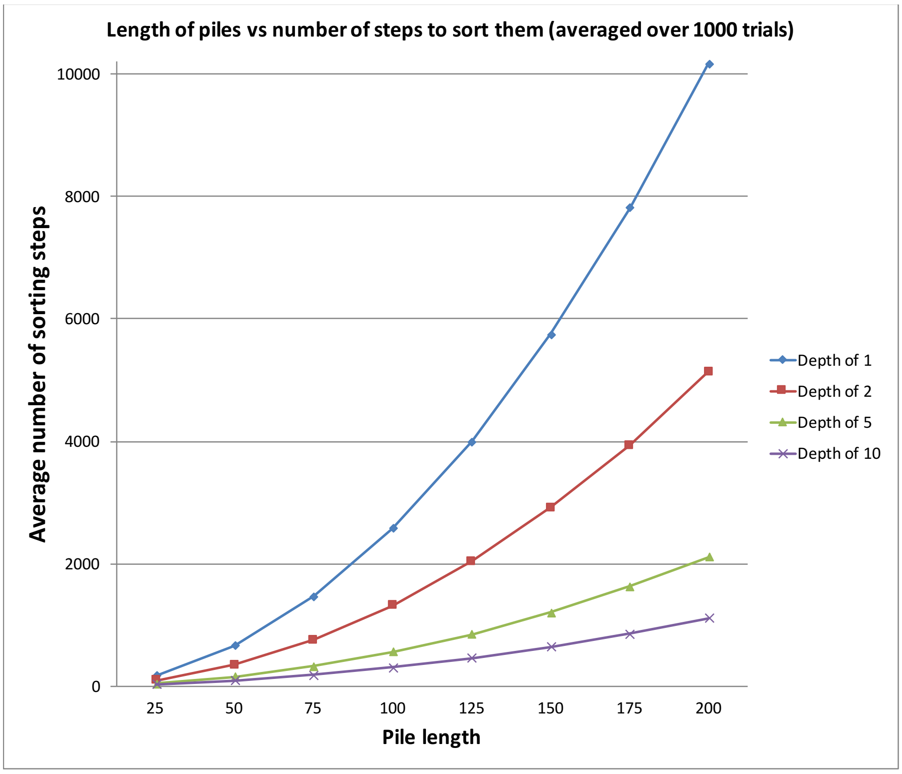
COSC 241 Assignment report

by Johnny Flame Lee and Frida Israelsson



To investigate the relationship between pile length, depth of access and the number of steps it takes to sort the pile, we conducted trials with a range of pile sizes (25, 50, 75… 200). In order to ensure the consistency of the experiment, the ordering of numbers within each pile was done by random permutation. For each experiment with a given pile length we repeat the process 1000 times, taking the averaged value as the result of the experiment to minimise bias.

One obvious pattern we have observed is that at the default depth of 1, the number of steps require to sort the pile grows rapidly as the input value increases, it became in practical to test pile sizes over 200 at the current number of repeats (1000).

We estimate the complexity of this search algorithm to be O(n^2). In the worst case scenario, for a pile size of 200 and a search depth of 1, it would take 200 times to find the desired value. The next iteration may take up to 199, 198 and so on. By increasing the size of the depth, we go about this problem in a “divide and conquer” fashion— break the problem into smaller chunks reducing the constant of n. A pile depth of 2 on a pile of length 200 would take 1/2 as long as using the depth of 1 on the same pile, while a depth of 10 would require 1/10 of the time.

It is to be noted, however, while divide and conquer can reduce the time taken to process a given pile compared to brute force (searching one item at a time). The overall complexity of the algorithm remains unchanged because we are only optimising the constant, which becomes less and less relevant as the the size of n increases.