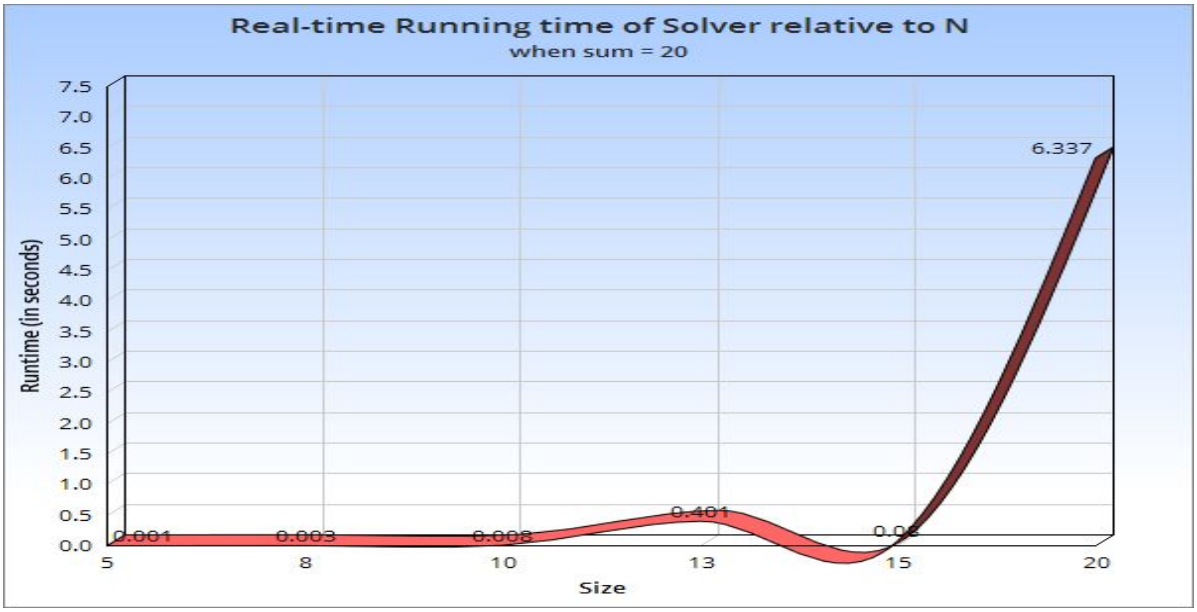
CHO, Guk Hum Jaeson
PONCE DE LEON, Allen O.
SAMARISTA, Rainier John P.

CMSC 142 C-2L

Subset Sum Solver Program using Backtracking Analysis

The program that the group created which uses backtracking has a growth rate that is undeniably worse compared to the dynamic programming of the Subset Sum Solver having a time complexity of only O(N*sum), significantly faster than the program the group has produced, having an exponential growth in its time complexity. The program, having only one solution would still lose by a certain amount against the refined dynamic code which we obtained from the link: https://www.geeksforgeeks.org/subset-sum-problem-dp-25/. This is because the backtracking method will still require the reiteration of the process in its worst case. The main cause of this drastic difference in time complexity is the tediousness of the backtracking technique, requiring the function to inefficiently check all possible combinations even if the same series of combinations has already been inspected in a previous iteration, unlike the usage of a boolean matrix in the dynamic programming version(one of a couple of approaches used for the problem) of this same code.



A variety of variables strongly affecting the running time of the program includes, but not limited to, is the machine's capabilities. Another that affects the program is the capacity of the machine's memory. A problem encountered by the group is the inability to run the program with an input size equal to, and exceeding 25, having a sum of 20. The cost of backtracking in regards to space is huge in this aspect compared to the space complexity of the dynamic programming's version of the code having an O(sum) space complexity (https://bit.ly/34NANyT). In regards to the program's input, both the input size and the sum affects the speed, but the size of the input weighs heavily compared to the sum of the input. In conclusion, using backtracking techniques will almost definitely yield a solution but would come at the expense of almost all resources involved in its computation.