COP 3502 Suggested Program Edits: Backtracking, Binary Search (Week 11 Programs)

1) To get a feel for the improvement backtracking yields, edit the 8 queens code, setting SIZE = 11 and get rid of the inner if statement which checks for a conflict. Run the code and see how long it takes. Now, put that if statement back in (the backtracking), and see how much faster it runs.

2) Change the digit divisibility problem definition so that the sum of the digits of each prefix of size k has to be divisible by k. Using this definition, something like 13263 is a digit divisible number since 1 is divisible by 1, 1 + 3 = 4 is divisible by 2, 1 + 3 + 2 = 6 is divisible by 3, 1 + 3 + 2 + 6 = 12 is divisible by 4 and 1 + 3 + 2 + 6 + 3 = 15 is divisible by 5. Rewrite the code so it prints out all values that satisfy this definition upto some threshold number of digits.

3) Both posted Sudoku solutions use a one dimensional array to store the 9 by 9 Sudoku board. Rewrite the code to work for a two dimensional integer array of size 9 by 9.

4) Rewrite the code for approach to work without the struct interval and just use two arrays - one to store all the start values and another to store all the end values.

5) Write your own square root function using binary search. For any positive real number x, the square root of it is always in between the interval 1 and x. (Note, if x is less than 1, it's the smaller end of the interval and if x is greater than 1, it's the larger end of the interval.)