CSCI-1200 Data Structures — Fall 2020
Lab 7 — Recursion

This lab gives you practice in the use of recursion to solve problems. All three checkpoints addressed in this lab deal with finding and counting the number of paths between points on a rectilinear grid. A starting point (x, y) with non-negative integer coordinates is provided. You are only allowed to move horizontally and vertically along the grid. Hence, from (x, y) you may move to $(x+1, y)$, $(x-1, y)$, $(x, y-1)$, $(x, y+1)$. Your goal is to return to the origin $(0, 0)$ in such a way that you **never** increase the distance to the origin. The distance is counted as the minimum number of total vertical and horizontal steps to reach the origin. In the first checkpoint the grid will be "free". In the second and third checkpoints, some of the grid locations will be "blocked" in the sense that you can not move to that point.

Stop now and play with small examples. Draw pictures of a grid. Think about the implications of the rules before you proceed with the checkpoints.

Checkpoint 1

estimate: 10-30 minutes

Did you notice that the rules prevent certain moves from occurring? What are they? If you don't get them right you will not be able to do the lab correctly. Confirm your understanding with one of the lab TAs.

Now, write a simple recursive program (from scratch) that reads a starting location, (x, y) and returns the total number of different paths back to the origin when the entire grid is "free". Two paths are different if there is at least one step on the path that is different even if most of the steps are the same. As a hint, when $x == 0$ and $y == 0$ there is 1 path, when $x == 2$ and $y == 1$ there are 3 paths, and when $x == 2$ and $y == 2$ there are 6 paths. When you're confident your program is debugged try even bigger values. For what values does the program take a few seconds to complete on your computer? If you increase these values by 1, how does it impact the running time? Is this what you expected from Big O Notation?

To complete this checkpoint show a TA your program to count *all paths* back to the origin. Be prepared to discuss the running time of your program for different input values.

Checkpoint 2

estimate: 20-40 minutes

Please download the files:
http://www.cs.rpi.edu/academics/courses/fall20/csci1200/labs/07_recursion/start.cpp
http://www.cs.rpi.edu/academics/courses/fall20/csci1200/labs/07_recursion/grid1.txt
http://www.cs.rpi.edu/academics/courses/fall20/csci1200/labs/07_recursion/grid2.txt
http://www.cs.rpi.edu/academics/courses/fall20/csci1200/labs/07_recursion/grid3.txt
http://www.cs.rpi.edu/academics/courses/fall20/csci1200/labs/07_recursion/grid4.txt

Starting from the supplied program, `start.cpp`, write a program to count the number of paths when some of the grid locations are blocked. When a location is blocked, no path can go through it. Before writing your own code, compile and run `start.cpp`. Use the files `grid1.txt`, etc. as input. These have a sequence of blocked locations, ending with the point location $(0, 0)$ (which isn't blocked, but signals the end of the input). The next location is the location for the initial (x, y). By changing this location you will be able to test your program in different ways.

To complete this checkpoint show a TA your code to find and count legal (unblocked) paths.

Checkpoint 3 will be available at the start of Wednesday's lab.
https://submitty.cs.rpi.edu/courses/f20/csci1200/course_materials