Project 2 – Dynamic Programming

COT 4400 – Fall 2017

Due 11/7/2017 by 5:00pm

1 Overview

You are required to solve a dynamic programming problem, and write a memoized version using either C++ or Java.

2 Description

Robots are trying to climb over a wall, however, due to some questionable programming choices the robots are bound by certain rules. To climb over the wall, they can only stand on the top of each other's shoulders, they can only stack so many on top of each other, and they can only create so many stacks. Given the number of stacks (*n*), and the max number of robots that can go in each stack (k) your task is to figure out how many way a given number of robots (*b*) can distribute themselves into the stacks. Your solution **must** use *dynamic programming*.

3 Report

You must include a report with your project. It will include the team members names along with their contribution to the project, programming language used (C++ or Java), detailed instructions on how to compile and run your program, and you must also answer the following questions.

a. What is the recurrence you are using for this problem?
b. What are the base cases of said recurrence?
c. What are the time and space complexities of your algorithm?
d. Write pseudo-code for an iterative approach to this algorithm. Your pseudo-code needs to be detailed and clear so that a programmer could implement it in their programming language of choice. **NOTE: You only need pseudo-code for the iterative approach. You don't need to write actual code.**
e. What are the time and space complexities for your iterative approach?

3 Code

You must implement the programming aspect of this project in either C++ or Java. The only rules for your program are that it must take a text file as input, and write

the results of the program to a text file as output. The name of the input and output files need to be given to the program as command line parameters. For example:
RobotStack input.txt output.txt

3.1 Input file format

The first line is a single integer that details the number of instances. Each line after the first has 3 comma separated values that detail an instance, and represent *b, n, k* in that order.

3.2 Output file format

Each line corresponds to an answer to an instance – total number of ways the robots can distribute themselves.

3.3 Example

Given *b=3, n=4, and k=2* there are 16 different ways the robots can distribute themselves. *HINT: this problem is* $\binom{n}{b}$ *when k=1 and* $\binom{n+b-1}{b}$ *when k=b. Note the solution increases as k move from 1 to b.*

The rest of how you code the solution is up to your team, however, as stated in section 3 you must give detailed instructions on how to compile and build your program. If the professor and/or TA are unable to compile and run your program from your instructions you will be docked a significant amount of points. As in project 1, your code must run on the C4 lab machines.

4 Submission

Submit a zip archive containing (1) all your code needed to compile and run your program; (2) the project report (PDF); and (3) a sample output file that your program created.

5 Grading

Algorithm implementation 70% each
Project report – 30%
NOTE: If you are on a team and do not contribute, you will not receive any points.