CSS342 Data Structures, Algorithms, and Discrete Mathematics I

Autumn 2017

Assignment 2: Recursion
Due date: Friday 27 Sep

Goal

To review the use in design and implementation of a recursive solution to solve a problem. Analyze the complexity of a recursive algorithm and practice the use of recurrences.

Overview of complex Class

Write a recursive program to solve the problem of the Fiji Explorers: three explorers and three cannibals come to a river and find a boat that holds two individuals. If the cannibals ever outnumber the explorers on either bank, the explorers will be eaten. How might they cross safely? Your task is to write a program that will show the sequence in which all the individuals are transported from one side of the river to the other side, well and alive.

Before starting to code, design and analyze your algorithm.

This is typically a backtracking problem (like the 8 queens). Understand what your search space looks like and identify valid/invalid states.

Hint: You can use a string to represent a state, i.e., showing the number of explorers and cannibals at both sides of the river/boat. Figure 1 below is an example of suggested output formatting. Your implementation doesn't have to look exactly the same -- this is just to give you some ideas of how you could present your output.

```
Sequence of Moves:
EEECCC \--/          ------
            \EC/
EECC--           \--/ EC----
            \E-/
EEECC- \--/          C-----
            \CC/
EEE---           \--/ CCC---
            \C-/
EEEC-- \--/          CC----
            \EE/
EC----           \--/ EECC--
            \EC/
EECC-- \--/          EC----
            \EE/
CC----           \--/ EEEC--
            \C-/
CCC--- \--/          EEE---
            \CC/
C-----           \--/ EEECC-
            \E-/
EC---- \--/          EECC--
            \EC/
------           \--/ EEECCC
```

Figure 1. Example of output: letter E represents and explorer and C a cannibal. A string surrounded by \ / is my attempt to represent the boat, and - symbols indicate empty placeholders for the humans in question.

Deliverables

1. C++ implementation (**source code**), as a file named `cannibals.cpp`. Make sure to properly comment your code and follow consistent style.

2. Documentation: Word or pdf **document** addressing the following ***design*** aspects
   a. Design: Describe your approach to solve this problem. Include your algorithm (pseudocode) and explain what is your base case and what the recursive step are. Why did you decide to approach the problem this way?
   b. Show and explain your output
   c. Compiling and running instructions (see deliverable 3 below).

3. Compiling and running your code: you must provide the compiling string (instructions) for us to compile your code on **Linux**.
   a. The compiling string must generate an executable file named `cannibals`

    b. Particularly, if you are using the STL or other standard libraries, include compiling instructions **both,** in your design document and as a comment at the top of your .cpp (and .h if any).

4. Submit all your files bundled as a zip file named

```
yourlastname:studentnumber_A2.zip
```

Grading Guide

[8pt] Design document

- Detailed description of approach [4]
- High level algorithm (use pseudocode -- not code!) [3]
- Compiling instructions [1pt]

Note: If you used external resources but you do not include them as references, the design document will receive zero points.

[12pt] Implementation

Correctness [10pt]

- Successful compilation [2 pt]
- Correct recursive implementation (base case, recursive calls, use of call stack) [4 pt]
- Correct Output [3pt]
- Explores all search space [1pt]

Note: if your program doesn't compile on a linux machine from the linux lab, it will receive zero points.

Program organization [2pt]

- Proper comments [1pt]
- Coding style [1pt]