

HW4: Recursive Puzzle

CSE2050 Fall 2017

Instructor: Jeffrey Meunier & Wei Wei

TA: Jenny Blessing, Param Bidja, Yamuna Rajan & Zigeng Wang

1. Introduction

In this exercise, we will solve a puzzle using recursion. Specifically, the puzzle involves traversals of a list.

2. Objectives

The purpose of this assignment is to help you:

- Sharpen your knowledge on recursion
- Refine your ability to translate real-world scenarios into code

***Note:** Before you start, if you are not familiar with recursion you are recommended to review the sample code we covered in lecture first.*

3. Background

You have been given a puzzle consisting of a row of squares each containing an integer, like this:

3	6	4	1	3	4	2	5	3	0
---	---	---	---	---	---	---	---	---	---

The circle on the initial square is a marker that can move to other squares along the row. At each step in the puzzle, you may move the marker the number of squares indicated by the integer in the square it currently occupies. The marker may move either left or right along the row but may not move past either end. For example, the only legal first move is to move the marker three squares to the right because there is no room to move three spaces to the left.

The goal of the puzzle is to move the marker to the 0 at the far end of the row. In this configuration, you can solve the puzzle by making the following set of moves:

Starting position	③	6	4	1	3	4	2	5	3	0
Step 1: Move right	3	6	4	①	3	4	2	5	3	0
Step 2: Move left	3	6	④	1	3	4	2	5	3	0
Step 3: Move right	3	6	4	1	3	4	②	5	3	0
Step 4: Move right	3	6	4	1	3	4	2	5	③	0
Step 5: Move left	3	6	4	1	3	④	2	5	3	0
Step 6: Move right	3	6	4	1	3	4	2	5	3	①

Even though this puzzle is solvable—and indeed has more than one solution—some puzzles of this form may be impossible, such as the following one:

③	1	2	3	0
---	---	---	---	---

In this puzzle, you can bounce between the two 3's, but you cannot reach any other squares.

4. Assignment

Write a recursive function that takes a **starting position of the marker** along with **the list representing the squares** and a **set of all the previously visited indices** (this set will always start off as empty). The function should return **True** if it is possible to solve the puzzle from the starting configuration and **False** if it is impossible.

You may assume all the integers in the list are positive (or zero) and the last entry, the goal square, is always zero. The values of the elements in the list must be the same after calling your function as they are beforehand, (which is to say if you change them during processing, you need to change them back!)

Additional Examples:

4	0	0	0	0
---	---	---	---	---

True - simply move right 4

1	2	1	3	2	0	0
---	---	---	---	---	---	---

True - move right 1, right 2, right 3

0	1	1	0
---	---	---	---

False - stuck at index 0

1	0	1	0
---	---	---	---

False - stuck at index 1

4.1 Approach

This problem should be solved recursively. Recursion involves calling a function within itself with input that is 'closer' to its base case. The first step is to define your base case (i.e. when have I solved the problem to the point where there are no more subproblems?). The next step

is to determine your recursive call - the key to this step is to ensure your recursive call is getting 'closer' to your base case to avoid infinite recursion.

Specifically for this puzzle, you'll need to keep track of which indices you've visited to avoid an infinite traversal (like the [3,1,2,3,0] example). **You should use a Set to keep track of where you've been, and only move on to check the next destination if it is not in that set and if that destination is in the bounds of the list.**

5. Submit your work to Mimir

Submit puzzle.py to Mimir after you complete your code. The Mimir will automatically grade your submission based on different unit tests. You can submit your code to Mimir up to **30 times** to refresh your existing score before the submission deadline.

6. Due date

This lab assignment is worth 4 points in the final grade. It will be due by 11:59pm on Mon. Oct 23rd 2017. A penalty of 10% per day will be deducted from your grade, starting at 12:00am.

7. Getting help

Start your project early, because you will probably not be able to get timely help in the last few hours before the assignment is due.

- Go to the office hours of instructors and TAs.
 - Prof. Wei Wei: Mon. 2:30 - 3:15pm, Wed. 2:30 - 3:15pm, Thurs. 2:30 - 3:15pm, Fri. 2:30 - 3:15pm @ITE258
 - Jenny Blessing: Fri. 12pm - 2pm @ITE140
 - Param Bidja: Tues. 2pm - 3pm @ITE140
 - Yamuna Rajan: Tues. 11am - 12pm, Wed. 9:30am – 10:30am @ITE140
 - Zigeng Wang: Mon. 3pm - 4pm @ITE140
- Post your questions on Piazza. TAs and many of your classmates may answer your questions.
- Search the answer of your unknown questions on the Internet. Many questions asked by you might have been asked and answered many times online already.