Lab10: The Dime-and-Penny Switcheroo

CSE2050 Fall 2017
Instructors: Jeffrey Meunier & Wei Wei
TAs: Jenny Blessing, Param Bidja, Yamuna Rajan & Zigeng Wang

1. Introduction

In this assignment, we use graph traversal to solve an interesting problem from the book "Perplexing Puzzles and Tantalizing Teasers" by Martin Gardner.

2. Objectives

The purpose of this assignment is to give you experience e in:

- Understanding the BFS graph traversal
- How to use BFS to solve some real world problems.
- Understanding when to use BFS and when to use DFS.

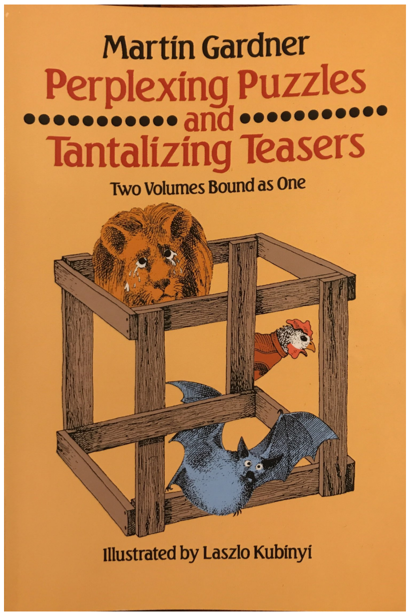
Note*: Before you start, if you are not familiar with graph and graph traversal algorithms, it is recommended you review the corresponding class notes.*

3. Background

3.1. The Dime-and-Penny Switcheroo

This problem is from book "Perplexing Puzzles and Tantalizing Teasers" by Martin Gardner. This problem is the 9th problem in the book.

The following page displays the cover of the book, and the illustration and description of this problem from the book.


Martin Gardner
Perplexing Puzzles and Tantalizing Teasers
Two Volumes Bound as One
Illustrated by Laszlo Kubinyi

9 *The Dime-and-Penny Switcheroo*

Put two dimes and two pennies in the spaces that contain their pictures. The object is to make the pennies and dimes change places in *exactly eight moves.*

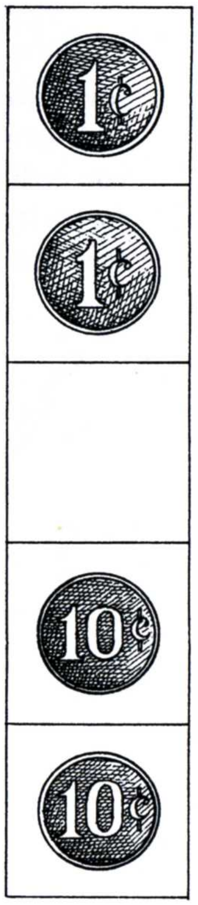
You are allowed two kinds of moves:

1. You can *slide* any coin into an empty space next to it.

2. You can *jump* any coin over the coin next to it, like a jump in checkers, provided you land on an empty space.

The puzzle isn't as easy as it looks. Time yourself to see how long it takes you to switch the coins in eight moves. If you solve it in five minutes, you're a genius. Ten minutes is excellent. Twenty minutes is about average.

Remember, only eight moves are allowed. If you do it in *more* moves, you haven't solved the puzzle.



We will solve this problem using graph traversal. We consider the positions of the 4 coins as one state, which is regarded as a vertex in the graph we try to construct. We use a 5-tuple to represent a state of the coins. For example, the state shown in the previous page is represented as (1, 1, 0, 10, 10). We use 0 to denote that the space at the position is empty.

Therefore, the starting vertex is (1, 1, 0, 10, 10) and the end vertex is (10, 10, 0, 1, 1). We need to use BFS to traverse from the starting vertex to the end vertex. We need to use BFS because we want to find the shortest path between these two vertices.

Before we can traverse this graph, we first need to construct the graph. We first obtain all the vertices for the graph. We get all the permutations of the 4 coins and an empty space and add all these states as vertices of the graph.

We then construct all the edges of the graph. We construct the edges for this graph as follows. We do a nested loop for vertex u and v, and decider whether (u, v) is an edge of the graph. If so, we add this edge to the set of the edges.

Once the graph is constructed, we then do the BFS traversal and print out the solution.

4. Assignment

In the skeleton zip file, you will find a skeleton for your .py file named *coins.py*. All you need to do is to complete the skeleton code based on the instructions and submit it to the Mimir system.

There are two places you need to fill in your code.

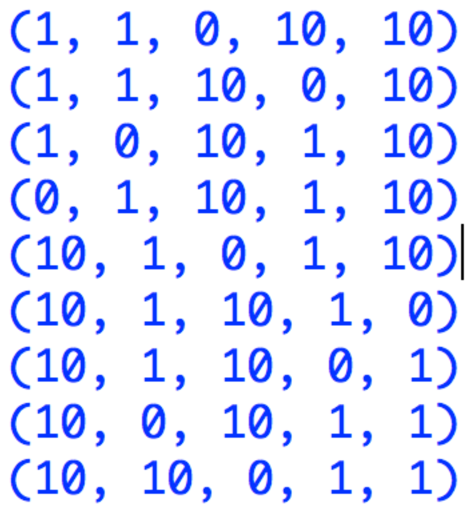
4.1. isneighbor(v1, v2)

Here we need to determine whether vertex v1 and v2 are neighbors based on the rules of two types of allowed moves. Note the method needs to return a Boolean value.

4.1. showresult(self, u, v)

This method does the BFS starting from vertex u. We use BFS in this case because we're interested in finding the shortest path – BFS visits vertices in order of distance so BFS can be used for the shortest path problem. We check whether the end vertex v is in the BFS tree. If so, we print out all the vertices traverse.

Below is a screenshot of my results.



```
(1, 1, 0, 10, 10)
(1, 1, 10, 0, 10)
(1, 0, 10, 1, 10)
(0, 1, 10, 1, 10)
(10, 1, 0, 1, 10)
(10, 1, 10, 1, 0)
(10, 1, 10, 0, 1)
(10, 0, 10, 1, 1)
(10, 10, 0, 1, 1)
```

5. Submit your work to Mimir

Submit your code to Mimir after you complete your code. Mimir will automatically check the output of your code and grade your submission. 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 **2 points** in the final grade. It will be due by **11:59pm on December 6th, 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.