

Introduction to Computers and Engineering Problem Solving

Spring 2012

Practice-only Problem Set 11: Red Balloon Experiment

Solutions posted Thu, May 17, 11 pm.

1. Problem Statement

In December 2009, DARPA ran a network challenge in the vein of the DARPA grand challenge (although for much less money). In this challenge, participants were tasked with finding 10 red weather balloons distributed throughout the continental US for 8 hours on December 5. This was a crowdsourcing activity, where people use social media tools to solve the problem.

The MIT Media Lab team won the challenge by creating a system where people received money not just for finding balloons, but also for getting people to join the hunt to find the balloons. Figure 1 shows an example of how the Media Lab compensated its participants. Carol found the balloon (for \$2,000), but Mark recruited Carol to look for it, so Mark got \$1,000. Colleen recruited Mark (and Chris and David), so she received \$500. Gene recruited James (who also recruited Colleen, but Gene was on the more direct path from Mary to Carol), who got \$250. Finally, Mary, who recruited Kathy and Gene, got \$125.

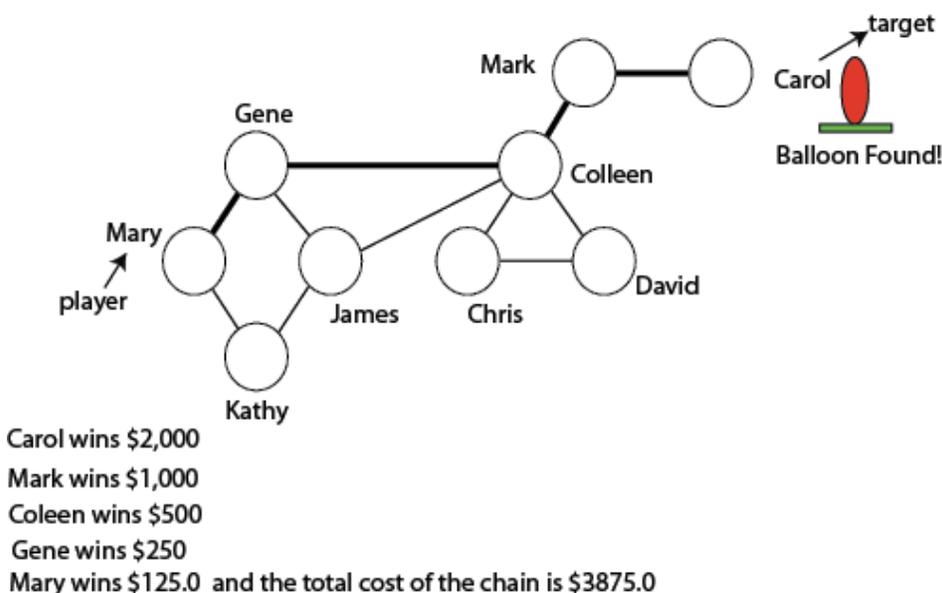


Figure 1: Chain of the Red Balloon experiment in a Social Network

The Media Lab team's procedure is simple:

1. One *player* signs up for the game and informs the connections in his/her social network. Each connection is a link in the network.

2. If someone in the network (target) finds the balloon, he/she receives a prize of \$2000.

3. Everyone in the shortest chain connecting the target with the player gets half of the next person's reward. Each person (name) is a node in the network.

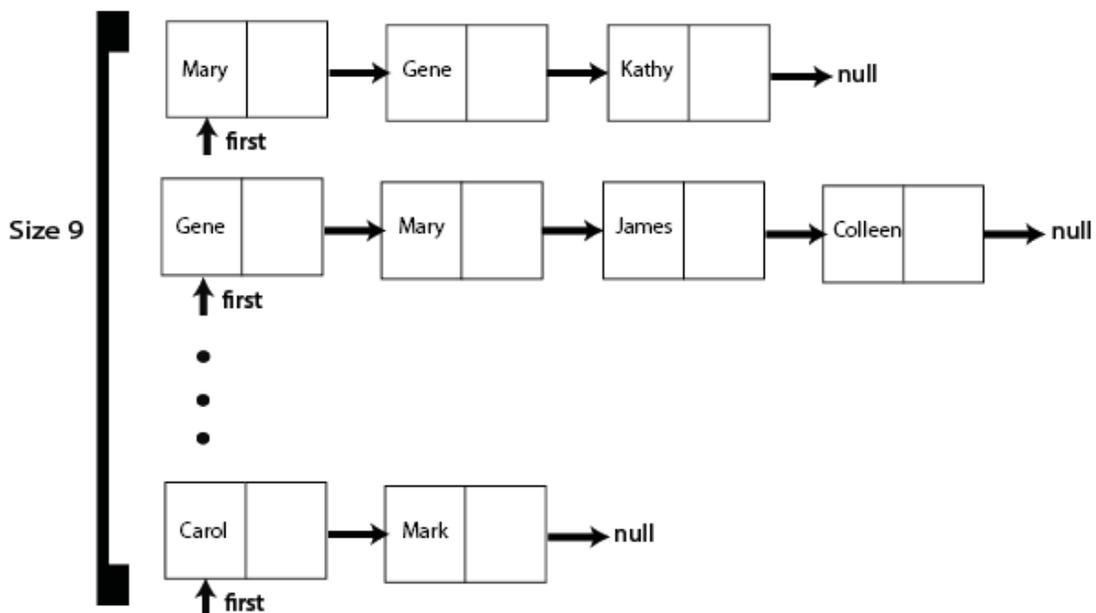
2. Program

To simplify this homework, we provide some detailed guidance on class and method design. Most of your code will be in class `SocialNetworkGame`.

a. In class `SocialNetworkGame`, create a graph that represents a social network, using the information in the file `graph.txt` that can be downloaded from the course Web site. The file contains:

- 9 names and their ID number, and
- 11 connections, listed as a pair of IDs. For each connection you must list the links between the two persons involved.

`graph` is an array of `SLinkedLists`, each element containing a person's name as the first item of the list followed by all of his/her connections (See Figure 2). With the input data in `graph.txt`, the graph will consist of 9 linked



lists.

Figure 2: graph

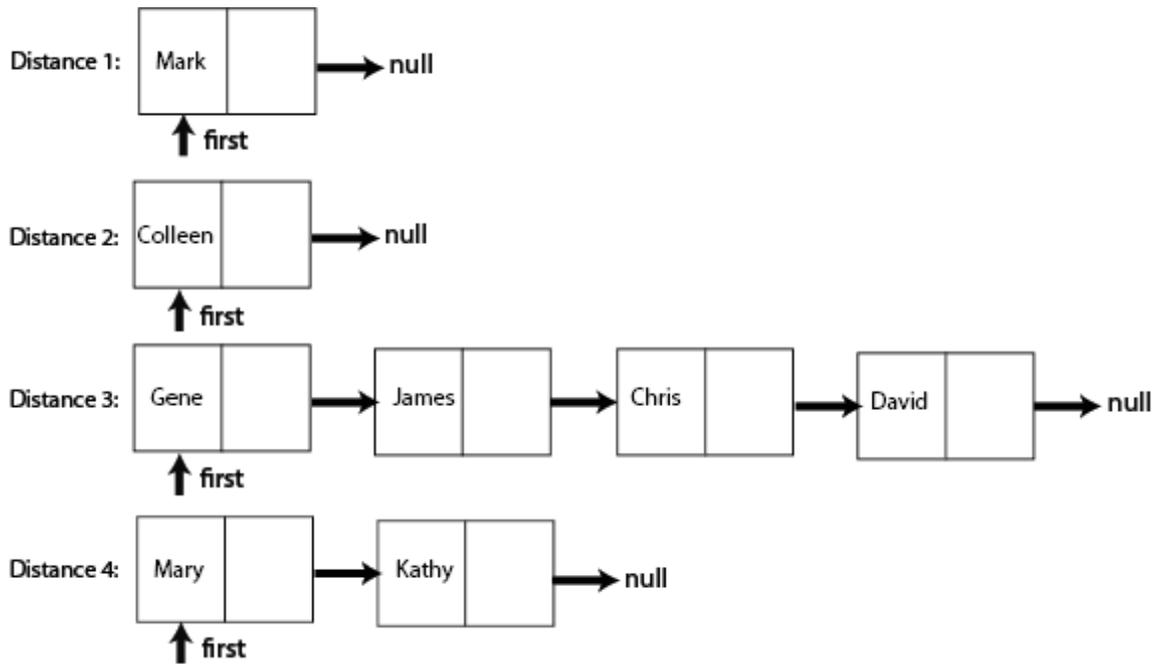


Figure 3: contacts

b. Still within class `SocialNetworkGame`, use graph to create `contacts`, which is an `ArrayList` of `SLinkedLists` that saves all the connections at each distance from the target to the player. In this example Carol is the target and Mary is the player. `contacts` is created saving the nodes (names) at each distance starting from the target node, and increasing the distance until the player is found.

c. Once `contacts` is built, you can determine the total cost of a chain from a target to a player, and the amount of money received by the player.

3. Provided Code

We provide `SLinkedList`, a copy of the class studied in lecture, with two additional methods, `addUnique()` and `getFirst()`. They are at the end of `SLinkedList`; they are at the end of the file and are very short.

A portion of `SocialNetworkGame` is provided. Key data member definitions are:

- `String player`: the name of the person who will report the red balloon.
- `String target`: the person who has found the red balloon.
- `double prize`: the amount of money to be given to the target person and to be proportionally distributed within the chain.
- `SLinkedList[] graph`: contains the index of a name followed by the index of the names of all of his/her connections

- `ArrayList<SLinkedList> contacts`: contains a list of nodes at a given distance from the target node. The distance is the number of links between two nodes. The direct contacts are at distance 1, the contacts of a contact are at distance 2, and so on.

We suggest you write the following methods; you are free use a different approach:

- `void setGraph(BufferedReader in)` This reads `graph.txt` and creates the social network in `graph`
- `int findNameIndex(String name)` This finds the index of a node (identified by its name) in the `graph` array. It's a useful helper method.
- `void setContacts()` creates the lists of contacts at a different distances between the target and the player in data member `contacts`
- `void calcDistances()`
- `void calcPrize()`

Create a test class with only a main method to:

- Create a `SocialNetworkGame` object
- Read in `graph.txt` and create the graph with `setGraph()`
- Build and output the lists of persons at different distances between the player and the target (`contacts`) with `setContacts()`
- Compute and output the total cost of the chain with `calcDistances()`
- Compute and output the amount received by the initial player with `calcPrize()`. You do not need to output the exact chain, but only the distance between target and player, which is enough to know the cost of the chain and the prize that the player receives.

Sample output for the data in Figure 1 and in `graph.txt`:

```
Carol and Mary are 4 levels apart
Carol's contacts at distance 1:
Mark
Carol's contacts at distance 2:
Colleen
Carol's contacts at distance 3:
James David Gene Chris
Carol's contacts of distance 4:
Kathy Mary
```

Mary wins \$125.0 and the total cost of the chain is \$3875.0

There is no turn-in for this practice-only homework.

MIT OpenCourseWare
<http://ocw.mit.edu>

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.