CS255, Spring 2014, SJSU
Homework 6

Fernando Lobo

Due: Apr 23, 2014

Problem 1

Consider the union-find data structure implemented with inverted trees as presented in class. Suppose you have 10 elements and the state of the *parent* and *rank* arrays are as follows,

index:	1	2	3	4	5	6	7	8	9	10
parent:	1	2	3	1	5	5	4	3	8	3
rank:	2	0	2	1	1	0	0	1	0	0

(Assume the indices of the *parent* and *rank* arrays correspond to the element labels. For example, the parent of node 1 is 1 and the rank of node 1 is 2.)

(a) Draw the tree representation for each set that is currently maintained by the data structure.

(b) Given a data structure with the state shown above, suppose you do Union$(7, 8)$ using the union-by-rank heuristic but without path compression. Redraw the resulting trees and show the contents of the *parent* and *rank* arrays after the operation is performed.

(c) Repeat the last question using both union-by-rank and path compression.

Problem 2

Exercise 22.1-1, page 592 from textbook.

Problem 3

Exercise 22.1-3, page 592 from textbook.

Problem 4

Exercise 22.2-1, page 601 from textbook.

Problem 5

Exercise 22.2-2, page 601 from textbook.

Problem 6

Suppose you a given a dictionary of n words of length 5 and you would like to find the minimum number of one-character changes that allows you do go from a starting word s to a target word t, subject to the constraint that whenever you change a character the resulting word has to be in the dictionary. As an example, here's a sequence of one-character changes that allows you to go from the starting word `coins` to the target word `money`:

```
coins
corns
cores
cones
coney
money
```

(a) Describe an efficient algorithm to solve this problem and argue that your algorithm is correct. There's no need to show pseudocode. You can explain your algorithm in plain english referring to algorithms we've seen in class, if needed. [Hint: think about each word as being a vertex in a graph.]

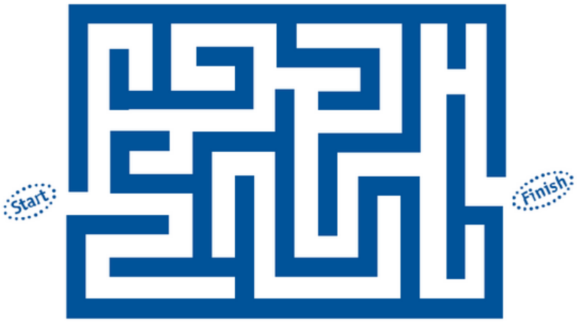
(b) What's the running time of your algorithm? Justify your answer.

Problem 7

Exercise 22.3-2, page 601 from textbook.

Problem 8

A maze can be modeled using an undirected graph by having a vertex for a starting point, a finishing point, dead ends, and locations in the maze where more than one path can be chosen, and then connecting the vertices according to the paths in the maze.



(a) Construct a graph for the maze shown above.

(b) Imagine you were lost in a maze and had access to a graph traversal algorithm to help you get out of it. Which algorithm would be more useful, DFS or BFS? Justify your answer. (Note that the maze could be very large and the corresponding graph could have a very large number of nodes.)