Yaxi Yu
V00828218

Csc226 assignment4 written

1. **These first problems are related to the code above.**

    a) **Explain what Create is doing. It is guaranteed to generate a maze that always has a unique solution. Why?**

    Create (int x, int y, int val) is actually building a random maze based on depth-first-traversal (DFS). Val is initialized as 0 so that at the beginning there isn't any wall that should be knocked down. But after that the value of val will decide which wall should be knocked down.
    It is guaranteed to generate a maze that always has a unique solution because in a tree, every tree edge is connected to another and there must be a unique path between them.

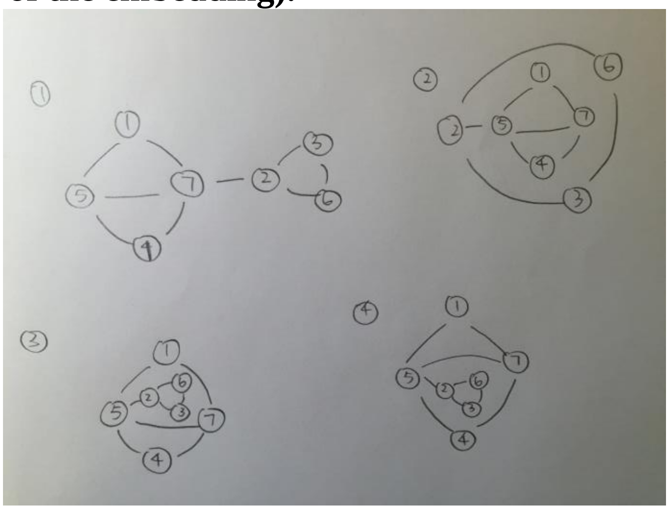
    b) **If there are n rows and m columns, then how many times is Create called when making a maze?**

    n*m times (because Create is called exactly once for each cell.)

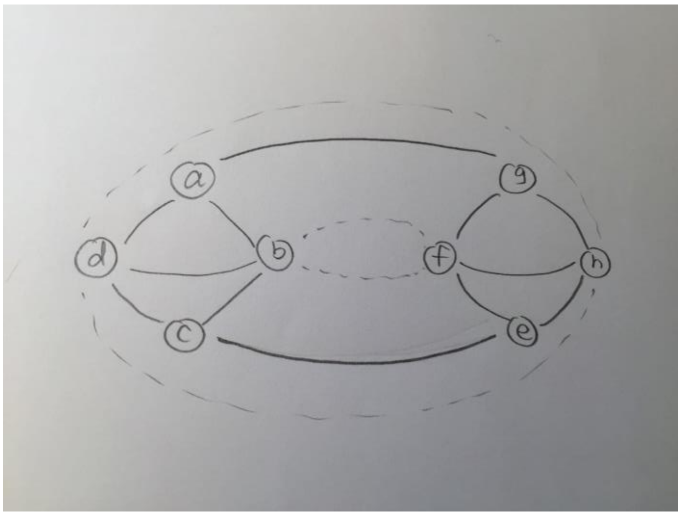
    c) **What is the purpose of the p^2?**

    P^2 is performed as P XOR 2. P^2 actually flip the direction. It always swap to the opposite wall of the cell. Like when trying to knock down left wall, you have to knock down right wall instead.

2. **Below is the rotation system of a plane graph. Draw all possible embedding in the plane. (i.e., each face should be an outer face in exactly one of the embedding).**



3. **Below is the rotation system for a graph embedded on a surface. What surface is it embedded on? Draw a nice diagram of the graph as embedded on this surface.**



Surfaces f=4, Vertices n=8, edges m=12
So g= (2-n+m-f)/2= 1
So it is embedded on a torus.

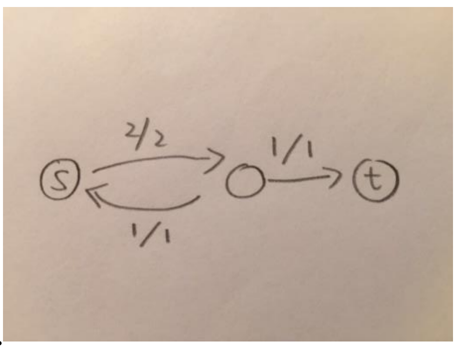
4. **From the book 4.4.17 (What happens if you allow a vertex to be enqueued more than once in the same pass in the Bellman-Ford algorithm?)**

The running time of the algorithm will go exponential.

5. **From the backtracking hangout: Question 2 about estimating the size of tree. Compare with the actual size of the tree (which can be obtained by running the program and counting the number of recursive calls).**

The estimated number of vertices is
1+8+8*6+8*6*4+8*6*4*3+8*6*4*3*2=1977.

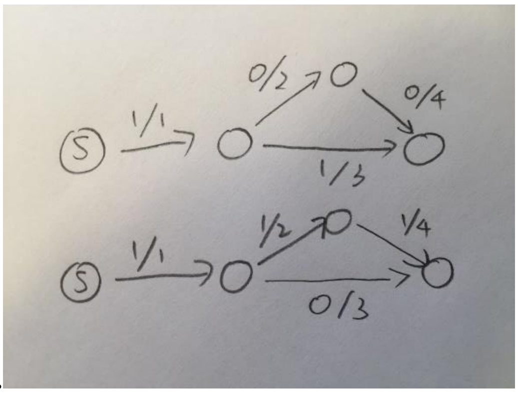
6. **From the book Exercise 6.38.**
   A) In any max flow, there is no directed cycle on which every edge carries positive flow.

False. 

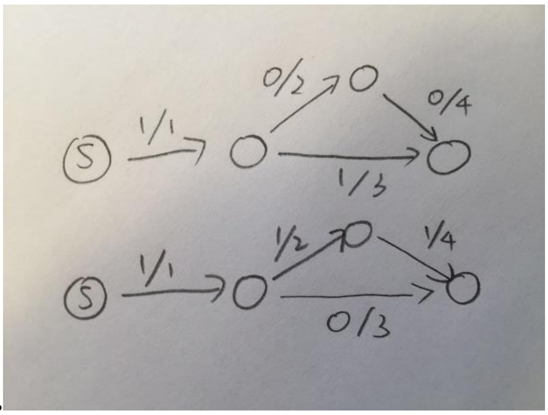
B) There exists a max flow for which there is no directed cycle on which every edge carries positive flow.

True.
Proof: if you reduce the flow of each edge in a positive flow cycle by the edge of minimum amount, the value of the flow is still maintain the same and one edge is set to zero.

C) If all edge capacities are distinct, the max flow is unique.

False.

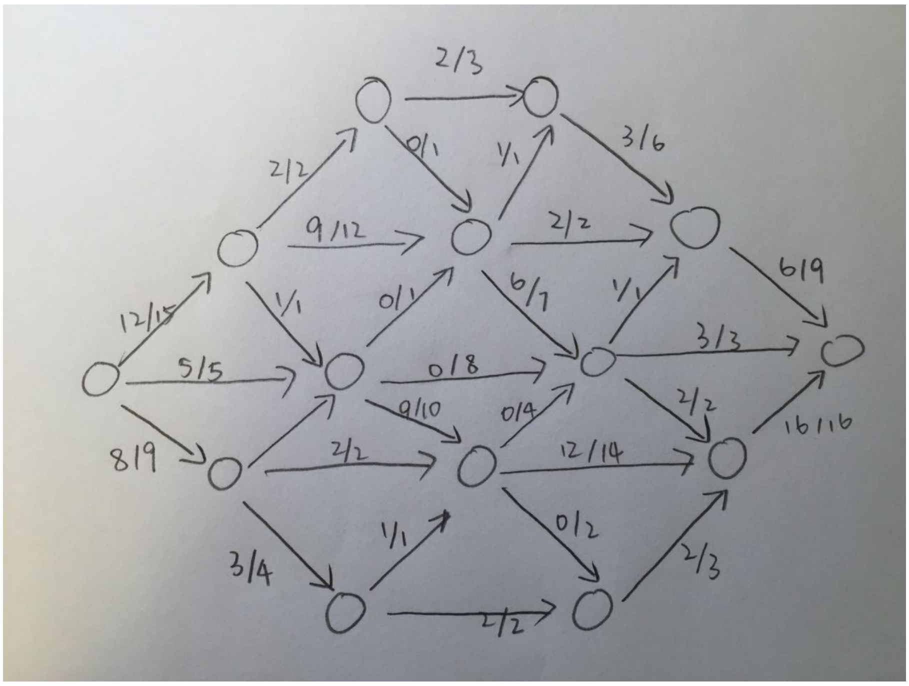
D) If all edge capacities are increased by an additive constant, the min cut remains unchanged.

False**.**

E) If all edge capacities are multiplied by a positive integer, the min cut remains unchanged.

True. Because every cut is multiplied by a constant, the relative order of cuts will not change.

7. Find the max-flow and min-cut in the attached network as it would be found by the Ford-Fulkerson algorithm.



The max flow is 12+5+8=25