CS 530: Algorithms --- Steve Homer--- Summer 2018

Homework 1 --- Due Wednesday, May 30

Reading : 1. Chapter 26, pages 708-736
2. Chapters 1-4, look over and read whatever seems new to you.

Problems:

1. (10 points). A dag is a directed acyclic graph (acyclic means no cycles). The topological sorting problem is: Given a dag G, sort the vertices of G is such a way that if G has an edge (u,v) then u occurs earlier than v in the sorted order.

One topological sorting algorithm given in our textbook (page 613) uses DFS as its main component. However there are simpler/more direct ways to do this.

i. Write an algorithm to give a topological sort of a dag G = (V,E) which does not use a depth first search.

ii. Carry out your algorithm on the graph in figure 22.8 on page 615 of our book.

iii. Show that your algorithm can be implemented in O(n+m) steps where n = |V| and m = |E|.

2. (12 Points) Answer the 4 T (true) or F (false) questions below.

Justify your answers here by saying why the statement is true or by giving a small counter example.

i. (True or False: Let (S,T) be a minimum s-t cut in the network flow graph G, and let (u,v) be an edge that crosses the cut in the forward direction, i.e., vertices s,u are in S and v,t in T.

If you increase the capacity of the edge (u,v) in G then you always also increase the maximum flow of G.

ii. True or False: If all capacities of a flow graph G are multiplied by some integer multiple of 5, then the values of the max flow of G is also a multiple of 5.

iii. True or false: There is no known polynomial time algorithm to solve the max-flow problem.

iv. True or False: Let G be a bipartite graph with n vertices and m edges. There
is an algorithm, to test if a matching for G is maximum which runs in time O(n+m) steps.

3. (10 points) Use the Ford Fulkerson algorithm to find the maximum flow in the following flow graph.

Find the min s-t cut of the graph as well.

5. (5 points) Give an example of a bipartite graph \( P = (L,R) \) whose corresponding flow graph \( G \) (as described at the beginning of Section 26.3 of the textbook) has a single unique max-flow and also the property that for every s-t min-cut \( (S,T) \) of \( G \) both \( S \) and \( T \) contains at least 2 vertices.

For the example you should draw the picture of \( P \) (showing \( R \) and \( L \)) and the corresponding picture of \( G \), and then write down all of the s-t min-cuts in \( G \) and as well as the one max flow of \( G \).

5. (10 points) (i). Recall the min cut problem for undirected graphs. Give an example of a graph \( G \) whose is min cut is not any single vertex of the graph. That is, for any vertex \( v \) in \( G \) the cut consisting of \( \{v\} \) on one side and all other vertices on the other is not a min cut.

(ii). Assume \( G \) is a 10 node graph and that \( G \) has a min cut of cost 5. Explain why \( G \) must have at least 25 edges.

(iii). Assume \( H \) is a 10 node graph which is a simple cycle. That is \( H \) is connected and every node in \( H \) has exactly 2 neighbors. Clearly the size of \( H \)'s min cut is 2. How many different min cuts does \( H \) have? Explain your reasoning.