CS420/520: Graph Theory with Applications to CS, Winter 2020

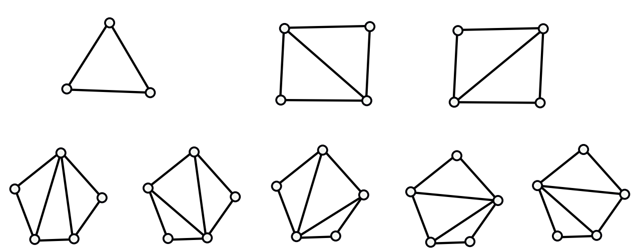
Homework 4

Due: Tue, 3/12/20

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. **Solutions must be typeset.**

Problem 1. Let C_n be the number of expressions containing n pairs of paranthesis that are correctly match. For example $C_1 = 1$: $()$; $C_2 = 2$: $()(), (())$; and $C_3 = 5$: $(()()), ((())), (())(), ()()(), ()(())$. Show that the number of triangulation of a convex polygon with $n + 2$ vertices is C_n. For example, the number of triangulations of convex polygons with 3, 4, and 5 vertices are 1, 2, and 5, as shown below.

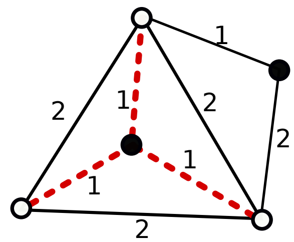


Problem 2. Consider the variant of the Metric TSP problem in which the object is to find a walk (which does not need to be closed) with given endpoints containing all the vertices of the graph.

(a) Design a 3-approximation algorithm for this problem.

(b) Design a 5/3-approximation algorithm for this problem.

Problem 3. The Steiner tree problem is as follows. Given $G = (V, E)$ with positive edge weights, and whose vertices are partitioned into two sets R (required) and S (Steiner), find a minimum cost tree in G that contains all required vertices. For the Steiner vertices you may decide to include or not include them. For example, in the figure below, the white vertices are required and the black vertices are Steiner. The red/dashed tree is the minimum Steiner tree.



Design a 2-approximation algorithm for the Steiner tree problem.