CS 630 - Fall 2020
Homework 5

Due: Thursday, December 10 by 12:00 midnight Eastern Standard (Boston) time
Note: This is a hard deadline. No HW will be accepted after December 10.

Reading : Read Chapter 35 on Approximation Algorithms, Sections 1, 2 and 3.

Problems:

1. One of the following three problems is known to be in NP.

Say which one is in NP and prove it.

i. Number of perfect matchings problem: The input is a bipartite graph G with the same number of vertices on each side of the graph. The output is the number of perfect matchings in G.

ii. The graph sub-isomorphism problem: Given two graphs G and H, does G have a subgraph which is isomorphic to H

iii. The clique problem: Given a graph G, what is the largest clique (that is complete subgraph) contained in G ?

2. (Traveling Salesman Problem) Do both part A and part B.
A: The problem here is to give an example of a complete weighted but NON-Euclidean graph G with 4 or 5 vertices which has the property that when you run the 2-approximation for the of the traveling salesman problem on the graph you obtain a TSP solution whose value is more than 20 times the optimal TSP solution for that graph.

(i). Draw your example weighted graph G, and state what makes it non-Euclidean.

(ii). Show what the optimal solution for G is, and if possible explain your answer.

(iii). Give the result of running the 2-approximation algorithm and show the steps you went through to get this approximation. (Show each step of the algorithm here, at least in brief.)

B: Give an example of a complete Euclidean graph on G with at least 4 vertices where the 2 approximation for the of the traveling salesman problem on the graph G yields a TSP solution whose value is more than 1.6 times the optimal TSP solution for that graph. Show enough of your work so that the validity of your example is clear.

3. (Bin packing) Do the first five parts of the problem 35-1 on page 1134 of our textbook. You don't need to do part f.

4. **Set Cover.** Recall the greedy set cover algorithm which was discussed in section and in our textbook. For this heuristic every step of the algorithm adds to the set cover the set in F which covers the most elements from U which are not yet covered by the algorithm.

(i). True or false: There is an instance I of the set cover problem on which the greedy set cover algorithm produces an answer which is $\geq$ OPT(I)+2. Explain your answer.
Note: Your explanation should either be an example showing the statement is true or a proof showing that it is not true.

(ii). Show that the following inequality is always true for the greedy set cover algorithm:
$|C| \leq |C^*| \times$ (the size of the largest set in $\mathcal{F}$), where C is set cover found by the greedy algorithm and C^* is the optimal set cover. Explain/justify your answer.