CSE 101 - ALGORITHMS - SUMMER 2000
Homework 3

Due Wednesday, August 2nd, 8am in the class. No exception!

Problem 1:

(10 points) We know that a graph $G = (V, E)$ can be tested for cycles using a modified depth-first search algorithm, which runs in time $O(|V| + |E|)$. You are asked to give an improved algorithm, running in time $O(|V|)$; your algorithm should return either YES or NO (do not return the cycle).

Problem 2:

(10 points) Dijkstra's algorithm finds the shortest paths in a weighted graph G from a vertex s to any other vertex in time $O(|V|^2)$. Now suppose that G is a weighted graph whose edges have weights either 1 or 2. Can you device a better algorithm? (Hint: You can find an algorithm that runs in time $O(|V| + |E|)$.)

Problem 3:

(10 points) The minimum spanning tree problem takes as input a connected undirected graph on n nodes whose edges have integer weights. Possible solutions are spanning trees of the graph, i.e., subsets of the edges that contain no cycles but connect all n nodes. The cost of a spanning tree is the sum of the weights of its edges, and the problem is to find a spanning tree of minimal cost.

Kruskal's algorithm is a greedy algorithm for this problem where, in every phase of the algorithm, a sub-forest of the tree has been chosen. In the phase, the minimum cost edge that connects two distinct components of this forest is added to the forest.

Prim's algorithm is also greedy but is slightly different. Here, at the beginning of a phase, a tree has been found that spans a subset V_T of the nodes. The minimum cost edge between a node in V_T and a node not in V_T is added to the tree, and the endpoint of that edge not already in is added to V_T.

1. (4 points) You are told that the inputs for your algorithm will all be planar graphs[1] and you know that planar graphs have at most $3n$ edges. Which algorithm do you pick? What is the overall time analysis?
2. (4 points) Say that you know, in addition, that all the edge weights of the graph are integers between 1 and n. Does this change which algorithm you choose? How you would implement the algorithm? Analyze your algorithm.
3. (2 points) Say that instead you want to find the *maximum* cost spanning tree. Can you use modified versions of these algorithms? Why or why not?

---

[1] A planar graph is a graph that can be drawn such that the edges do not intersect. Maps can be represented as planar graphs. However, you do not need to be familiar with planar graphs to solve this problem.