CS302 - Problem Set 10

1. In this problem set, you will consider another approach for mitigating harmful effects of algorithms: banning (through legislation) the use of certain types of algorithms, particularly in certain contexts. Read this webpage by Amnesty International advocating for the banning of facial recognition algorithms. Briefly research legislation pertaining to facial recognition algorithms here in Vermont.

   Do you think this is an good method of limiting the harm that algorithms can do? What are the advantages and disadvantages of this approach? Who gets to make decisions regarding these bans? Who can influence the people making these decision?

2. Consider the following variations on problems we've previously solved using dynamic programming algorithms. If the algorithm we used no longer works, explain why, and then briefly describe a fix and the impact of the fix on the runtime. If the algorithm can stay the same, you don't need to explain.

   (a) In MWIS on a line, we only considered graphs where the vertex weights were positive integers. Would we have to change the algorithm if we used negative integers instead?

   (b) In MWIS on a line, we only considered graphs where the vertex weights were positive integers. Would we have to change the algorithm if we used positive real numbers instead?

   (c) In the Knapsack problem, we only considered items whose values were positive integers. Would we have to change the algorithm if the values of items were allowed to be negative integers instead? (Assume that each item's cost is still a positive integer.)

   (d) In the Knapsack problem, we only considered items whose values were positive integers. Would we have to change the algorithm if the values of items were allowed to be positive real numbers instead? (Assume that each item's cost is still a positive integer.)

   (e) In the Knapsack problem, we only considered items whose costs were positive integers and the total capacity was a positive integer. Would we have to change the algorithm if the costs of items were allowed to be positive or negative integers? (A negative cost would mean that adding that item to your knapsack would increase the size of your knapsack.) (Assume that each item's value is still a positive integer.)

   (f) **[Challenge]** In the Knapsack problem, we only considered items whose costs and values were positive integers and the total capacity was a positive integer. Would

we have to change the algorithm if we used both negative and positive integers for item costs and values?

3. (a) In the version of Bellman-Ford we looked at in class, we assume that there are no negative cycles in the graph. Describe briefly in words how you could use the Bellman-Ford algorithm to detect whether there is a negative cycle or not (that is reachable via a path from s). Then modify our shortest path algorithm from class to become a negative-cycle-detection algorithm. Assume the G is given as an adjacency matrix w.

   (b) Next create an algorithm that, given a graph that contains a negative cycle reachable from s, will return the set of edges on the negative cycle. Assume the G is given as an adjacency matrix w. (This algorithm is used in currency arbitrage.)

4. Prove DOUBLE-3-SAT is NP-Complete. Recall DOUBLE-3-SAT: a yes instance is a description of an AND-of-ORs formula with n variables, where each clause involves at most 3 of the variables $u_1, u_2, \ldots, u_n$ (and their negations) and there are at least two *different* satisfying assignments to the formula. Otherwise output NO. For example, $x = (u_1 \vee \neg u_1 \vee \neg u_2) \wedge (u_2 \vee u_3) \wedge (\neg u_3)$ is a YES instance because it has two valid assignments, $u_1 = T, u_2 = T, u_3 = F$ and $u_1 = F, u_2 = T, u_3 = F$ but $x = u_1$ is a NO instance because there is only one satisfying assignment: $u_1 = T$.