CS 6150: HW3

Due Date:

This assignment has 5 questions, for a total of 100 points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question	Points	Score
Minimum-cost Tree	20	
Max-SAT Problem	20	
Containers and Truck	20	
Grid Graph	20	
Graph Coloring	20	
Total:	100	

Question 1: Minimum-cost Tree .... **[20]**

In this problem, the input consists of a complete graph $G = (V, E)$ with distances between all pairs of vertices, and a set $V' \subseteq V$. Suppose the distances in the input is a metric, and the weight of an edge connecting two vertices in G is defined as the distance between these two vertices.

(a) [**20**] Design a ratio-2 approximation algorithm to find a minimum-cost tree that includes V'. The cost of a tree is the sum of all weights of its edges. This tree may or may not include vertices in $V - V'$. Show the approximation ratio of your algorithm is 2. (Hint: Recall the approximation algorithm for the TSP.)

Question 2: Max-SAT Problem .... **[20]**

Suppose you are given a set of clauses, and each clause is the the disjunction of several literals. Your goal is to find an assignment that satisfies as many of these clauses as possible.

(a) [**8**] Here is a simple algorithm:

---

for each variable **do**
  set its value to either 0 or 1 by flipping a coin
end for

Suppose the input has m clauses, of which the jth has k_j literals, show that the expected number of clauses satisfied by the above algorithm is no less than $\frac{m}{2}$. In other words, this is a 2-approximation in expectation.

(b) [**12**] Improve the above algorithm to make it deterministic.

Question 3: Containers and Truck .... **[20]**

Suppose a ship arrives, with n containers of weight $w_1, w_2, \cdots, w_n$. Standing on the dock is a set of trucks, each of which can hold K unites of weight. (You can assume that K and each w_i is an integer.) You can stack multiple containers in each truck, subject to the weight restriction of K; the goal is to minimize the number of trucks that are needed in order to carry all the containers.

A greedy algorithm you might use for this is the following. Start with an empty truck, and begin piling containers $1, 2, 3, \ldots$ into it until you get to a container that would overflow the weight limit. Now declare this truck "loaded" and send it off; then continue the progress with a fresh truck. This algorithm, by considering trucks one at a time, may not achieve the most efficient way to pack the full set of containers into an available collection of trucks.

(a) [**5**] Give an example of a set of weights, and a value of K, where this algorithm does not use the minimum possible number of trucks.

(b) [**15**] Show, however, that the number of trucks used by this algorithm is within a factor of 2 of the minimum possible number, for any set of weights and any value K.

Question 4: Grid Graph .... **[20]**

Suppose you are given an $n \times n$ grid graph G as in the following figure.

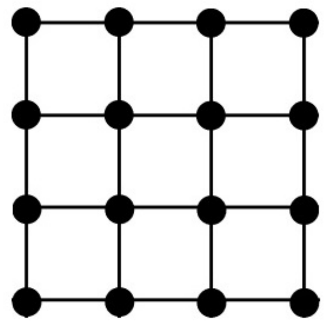

Figure 1: A grid graph.

Associated with each node v is a weight $w(v)$, which is a nonnegative integer. You may assume that the weights of all nodes are distinct. Your goal is to choose an independent set S of nodes of the grids, so that the sum of the weights of the nodes on S is as large as possible. (The sum of the weights of the nodes in S will be called its total weight.)

Consider the following greedy algorithm for this problem.

Algorithm 1 The "heaviest-first" greedy algorithm:

```
Start with S equal to the empty set
while some node remains in G do
  Pick a node v_i of maximum weight
  Add v_i to S
  Delete v_i and its neighbors from G
end while
return S
```

(a) [**7**] Let S be the independent set returned by the "heaviest-first" greedy algorithm, and let T be any other independent set in G. Show that, for each node $v \in T$, either $v \in S$, or there is a node $v' \in S$ so that $w(v) \leq w(v')$ and (v, v') is an edge of G.

(b) [**13**] Show that the "heaviest-first" greedy algorithm returns an independent set of total weight at least $\frac{1}{4}$ times the maximum total weight weight of any independent set in the grid graph G.

Question 5: Graph Coloring .... [**20**]

Solve Question 8, parts (a), (b) **ONLY** from `http://web.engr.illinois.edu/~jeffe/teaching/algorithms/notes/31-approx.pdf`. Each part is worth 10 points.

BONUS: for 10 extra points, solve Question 8(c). Note that you will solve this by constructing an example graph of size n and showing that you need $\omega(1)$ colors.