CS 180: Introduction to Algorithms and Complexity
Final Exam

March 17, 2021

Name	
UID	
Section	

1	2	3	4	5	6	Total

★ **Print your name, UID and section number in the boxes above, and print your name at the top of every page**.

★ **Your Exams need to be uploaded in Gradescope. Print your answers, or use a dark pen or pencil. Handwriting should be clear and legible.**

- Do not write code using C or some programming language. Use English in bullet points or clear and simple pseudo-code. Explain the idea of your algorithm and why it works.
- Your answers are supposed to be in a simple and understandable manner. Sloppy answers are expected to receive fewer points.
- Don't spend too much time on any single problem. If you get stuck, move on to something else and come back later.

1. **(a.)** Consider the problem of "Approx-3SAT": The input is the same as 3-SAT, which is a boolean expression $C_1 \wedge C_2 \wedge \cdots \wedge C_n$ where each C_i is an "or" of three literals (each literal can be one of $x_1, \ldots, x_m, \neg x_1, \ldots, \neg x_m$). Note that we assume a clause cannot contain duplicate literals (e.g., $(x_1 \vee x_1 \vee x_2)$ is not allowed). Instead of determining whether there's a truth assignment on $\{x_i\}_{i=1}^m$ that satisfies this boolean expression (which means it satisfies all the clauses), now we want to determine whether there's an assignment that satisfies at least $n-1$ clauses.
Prove that Approx-3SAT is NP-Complete.

   (b.) Give poly time reduction of Hamiltonian Cycle in an undirected graph to a Hamiltonian path in an undirected graph. I.e. if HAM path is poly then HAM cycle is poly. (Hint: Replicate some node twice, and now add "something" so that a HAM path in the new graph will not be in the "middle" of the path, but toward the "end." Alternatively, use cook reduction which allows to invoke HAM path $|E|$ times.)

   (c.) Although Vertex Cover (V.C.) is NP-Complete we want to show we can get a vertex cover that is at most greater by a factor of 2 than the size of the optimal V.C. in poly time.

   - A Maximal size (number of edges) Matching in an undirected graph G is a Matching M such that no edge e can be added to it such that $M \cup \{e\}$ is a matching in G. Give a linear time algorithm for for Maximal size matching assuming G is given by adjacency list.
   - Argue that the vertices which are adjacent on the edges in a Maximal size Matching constitute a vertex cover.
   - Argue that the optimal size V.C. is larger equal to the size of a maximal size matching.

2. There is an array with n integers, but the values are hidden to us. Our goal is to partition the elements into groups based on their values — elements in the same group should have the same value, while elements in different groups have different values. The values are hidden to us, but we can probe the array in the following way: we can query a subset of these n elements, and get the number of unique integers in this subset. Design an algorithm to partition these n elements in $O(n \log n)$ queries. (Hint: Assume inductively that you have solved the problem over the elements observed so far. Consider how to "observe" another element.)

3. **(a.)** Prove that Euclidean version of the Traveling Salesmen Problem (TSP) is NP-Complete. Euclidean TSP is a special case of general TSP, where distances in the complete graph obey the triangle inequality, i.e for each three edges that constitute a triangle the sum the weights of any two, is greater equal than the weight of the third edge.

   (hint: You can use the same kind of reduction (tweaked) we used in class to prove that the general TSP is NP-Complete.)

   (b.) Given a weighted complete graph $G = (V, E, w)$ with positive weights. Prove that the weight of an MST of G is lower equal than the optimal value of a TSP in G.

   (c.) Consider a pre-order (e.g. a DFS that allocate the next number upon the discovery of a new node) enumeration of the nodes of an MST of Euclidean G starting at an arbitrary node. We could use that sequence of nodes to traverse our graph G, i.e we traverse in the order $1 \longrightarrow 2 \ldots \longrightarrow n \longrightarrow 1$. Show that the total value of this traversal is less then 2*opt(TSP) solution.

   (d.) Prove that the size of the set of odd degree nodes S in an MST in G is even.

   (e.) There exists a polynomial time algorithm for finding minimum weight perfect matching in a complete weighted graph $G' = (V', E'.w')$ where $|V|$ is even.

   Consider the minimum weight matching in the weighted graph G' where G' is the graph induced by the weighted graph G over the set of nodes S. Argue that the total value of this minimum weight matching is at most $\frac{1}{2}*$opt(TSP) of G.

   (f.) Consider the bag of edges that contains the edges of an MST of the euclidean G plus the edges of the minimum weight perfect matching of S nodes of this MST. Show that three exists a traversal of this set of edges where every edge in the bag is traversed only once (if the bag contains two copies of the "same" edge then this edge is traversed twice).

   Also. prove that the sum of the the weights of the edges in the bag is at most $1.5*$opt(TSP).

   (g.) Using pre-order as before find a TSP whose length is at most 1.5opt(TSP)

4. **(a.)** CS students participate in many recognized Student Groups e.g. ACM-W, UPE, etc.. A student may be a member of many student organizations. Once a year the Department holds an advocacy session in which a single member of a group is advocating for an increase in the budget of a group. But, to avoid seeing same face for too long, a member can only advocate for a single group among the groups the student belongs to.

   We want to find whether each group can choose a unique member to represent it.

   E.g. If we have two singelton groups containing the same student then obviously the problem has no solution.

   Given groups and their membership determine whether choosing unique member from each group is NP-Complete problem or solvable in polynomial time. Either prove the NP-Completeness, or give a Poly-time algorithm.

   (b.) Given an undirected connected graph $G = (V, E)$,

   - A cut is the set of edges connecting nodes in different part in a 2-partition of the set of nodes of the graph. A cut is minimal, if no subset of it is a cut. Argue that the two parts corresponding to a minimal cut, each is connected subgraph of G.
   - Give an algorithm to find a minimum-cut $\textit{min-cut}_{a,b}$. A min-cut $\textit{min-cut}_{a,b}$ is a min-cut of G under the constraint that two given nodes $a \in V, b \in V$ are in two different partitions induced by the min cut.
   - Give a polynomial time reduction (Cook reduction) from min-cut of G to $\textit{min-cut}_{a,b}$ for various a and b. Analyze the number of times $\textit{min-cut}_{a,b}$ was called.
   - Given a partition A, B of the nodes $v_0, v_1, \ldots v_{n-1}$ of G, prove that there exist a number $i \in \{0, 1 \ldots . n-1\}$ such that $v_i \in A$ and $v_{(i+1) \mod n} \in B$.
   - Describe an algorithm the finds min-cut of G that calls on $\textit{min-cut}_{a,b}$ at most n times.