HW 11

Due on **Wednesday, December 4, 2024 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2024 Version: **1.2**

Submission instructions as in previous homeworks.

21 (100 PTS.) The End Is Nigh.

The following question is long, but not very hard, and is intended to make sure you understand the following problems, and the basic concepts needed for proving NP-Completeness.

All graphs in the following have n vertices and m edges.

For each of the following problems, you are given an instance of the problem of size n. Imagine that the answer to this given instance is "yes", and that you need to convince somebody that indeed the answer to the given instance is `yes`. To this end, describe:

(I) An algorithm for solving the given instance (not necessarily efficient). What is the running time of your algorithm?

(II) The format of the proof that the instance is correct.

(III) A bound on the length of the proof (its have to be of polynomial length in the input size).

(IV) An efficient algorithm (as fast as possible [it has to be polynomial time]) for verifying, given the instance and the proof, that indeed the given instance is indeed `yes`. What is the running time of your algorithm?

(EXAMPLE)

Shortest Path

Instance: A weighted undirected graph G, vertices s and t and a threshold w.
Question: Is there a path between s and t in G of length at most w?

Solution:

(I) **Algorithm**: We seen in class the Dijkstra algorithm for solving the shortest path problem in $O(n \log n + m) = O(n^2)$ time. Given the shortest path, we can just compare its price to w, and return yes/no accordingly.

(II) **Certificate**: A "proof" in this case would be a path π in G (i.e., a sequence of at most n vertices) connecting s to t, such that its total weight is at most w.

(III) **Certificate length**: The proof here is a list of $O(n)$ vertices, and can be encoded as a list of $O(n)$ integers. As such, its length is $O(n)$.

(IV) **Verification algorithm**: The verification algorithm for the given solution/proof, would verify that all the edges in the path are indeed in the graph, the path starts at s and ends at t, and that the total weight of the edges of the path is at most w. The proof has length $O(n)$ in this case, and the verification algorithm runs in $O(n^2)$ time, if we assume the graph is given to us using adjacency lists representation.

21.A. (20 PTS.)

Friendly Set

Instance: An undirected graph G, an integer k. **Question:** Is there a friendly set in G of size k?

For a vertex $u \in \mathsf{V}(\mathsf{G})$, let $\Gamma(u, \ell) = \{v \in \mathsf{V}(\mathsf{G}) \mid d(u, v) \leq \ell\}$ be the set of vertices in distance at most ℓ from u (here, $d(u, v)$ is the number of edges in the shortest path between u and v in G). The set X is ***friendly*** if for all $v \in \mathsf{V}(\mathsf{G})$, we have that $|\Gamma(v, 7) \cap X| < 12$.

21.B. (20 PTS.)

No Squares, Please

Instance: An undirected graph G with n vertices and m edges, a parameter k. **Question:** Is there a subset S of k vertices in the graph, such that no four vertices of S are included in a cycle of length four in G?

21.C. (20 PTS.)

StrangerThings.

Instance: S: Set of $2n$ positive integers. t, m: Integer numbers. **Question:** Are there disjoint sequences (each of m distinct numbers) $X = x_1, x_2, \ldots, x_m$, and $Y = y_1, y_2, \ldots, y_m$ of numbers, all taken from S, such that $\sum_{i=1}^{m} \lvert x_i - y_i \rvert = t$

21.D. (20 PTS.)

Sunflower

Instance: X a set of n elements, $\mathcal{F} = \{f_i \subseteq X \mid i = 1, \ldots, m\}$, and parameters α and β. **Question:** Is there a subset $S = \{s_1, \ldots, s_\alpha\} \subseteq \mathcal{F}$ of α distinct sets, such that they all share β common elements? Formally, $\lvert \cap_{i=1}^{\alpha} s_i \rvert = \beta$.

(Strangely, people[1] are interested in closely related problems, see `https://en.wikipedia.org/wiki/Sunflower_(mathematics)`.)

21.E. (20 PTS.)

SHALLOW COVER

Instance: $(\mathsf{U}, \mathcal{F}, k, d)$: U: A set of n elements $\mathcal{F}$: A family of m subsets of U, s.t. $\bigcup_{X \in \mathcal{F}} X = \mathsf{U}$. k, Δ: Positive integers. **Question:** Are there k distinct sets $S_1, \ldots, S_k \in \mathcal{F}$ that are Δ-shallow (i.e., no element is contained in strictly more than Δ of the chosen sets).

Formally, the sets $S_1, \ldots, S_k$ (together) are ***Δ-shallow*** if for all elements $x \in \mathsf{U}$, the depth

$$\mathsf{d}(x) = \lvert \{i \mid x \in S_i\} \rvert$$

is at most Δ.

[1] Okay, mathematicians

22 (100 PTS.) The Matrix Transmogrification.

Let $\mathcal{B}(2n, n)$ be the set of all binary strings (i.e., vectors) $v = (v_1, \ldots, v_{2n}) \in \{0,1\}^{2n}$, such that for all $i \in [\![n]\!]$, we have that $v_{2i-1} \neq v_{2i}$.

Matrixity

Instance: A binary matrix M of size $m \times 2n$ (i.e., m rows and $2n$ columns), where every entry is either 0 or 1.
Question: Is there a (column) vector $v \in \mathcal{B}(2n, n)$ such that $Mv = (1, 1, \ldots, 1)$. Here, the calculations are done over boolean logic, so multiplication is the $\wedge$ (i.e., and) operation, and addition is the $\vee$ (i.e., or) operation.

22.A. (20 PTS.) Prove that Matrixity is in NP.

22.B. (40 PTS.) Show a polynomial time reduction from Matrixity to SAT. (You need to prove your reduction is correct.)

22.C. (40 PTS.) Show a polynomial time reduction from SAT to Matrixity. (You need to prove your reduction is correct.)