CS3230: Design and Analysis of Algorithms
Semester 2, 2022-23, School of Computing, NUS

Written Assignment 3

Deadline: 14th April, 2023 (Friday), 11:59 pm

Instructions

- Submit the soft copy at Canvas → Assignment → Written Assignment 3.
- Write legibly. If we cannot read what you write, we cannot give points. In case you CANNOT write legibly, please type out your answers.
- When you submit your answer, please try to make it concise. However, it should contain all the relevant details. **The page limit is 4**.
- All the proofs must be written formally; no marks will be given for hand-waving arguments.
- Before submitting your answers, please make sure you rename your submission file in the following way: <Student ID>.pdf. (Note, your student ID is that starts with "A".)
- If you need any clarification on any of the questions, we strongly encourage you to post in Canvas Forum (instead of emailing us) so that everybody can see our responses. You may also approach Steven, Diptarka or your tutor.
- While proving NP-hardness, you may show a reduction from any of the NP-complete problems introduced in the lectures/tutorials/practice set, more specifically, Circuit Satisfiability, CNF-SAT, 3-SAT, Vertex Cover, Independent Set, Max-Clique, Hamiltonian Cycle, Traveling Sales Person, Subset Sum, Knapsack Problem.
- More hints will be released on 5th April (Wednesday), after 05:00 pm.

Question 1 [20 marks]: Given a connected undirected graph G, a *spanning tree* is a connected subgraph that is a tree and includes all the vertices of G. Given any undirected graph, we all know how to compute a spanning tree (we can even compute a minimum spanning tree for weighted undirected graphs, but for those who can't recall that algorithm, that is fine, we do not need it for this question). Now, let us consider the following variant of the spanning tree problem.

Given a connected undirected graph G, the problem is to decide whether there exists a spanning tree with each node having degree at most five.

Your task is to show that the above problem is NP-complete. For that purpose, show the following.

(a) **[5 marks]** Show that the above problem is in NP.

(b) **[15 marks]** Prove that the above problem is NP-hard. (**Hints:** Try a reduction from the undirected Hamiltonian cycle problem.)

Question 2 [20 marks]: The presence of missing values in the data objects is ubiquitous in practice. Missing entries could be due to various types of faults during data collection, data transfer, or data cleaning phase and may occur arbitrarily. Dealing with a data set with missing values is one of the infamous challenges in data science. The problems of filling up missing entries are often cast as optimization tasks. One such problem is the following:

You are given an $m \times n$ matrix M with each entry set to either 0 or 1 or a wildcard $*$. Furthermore, all the entries of the last (i.e., m^{th}) row of M are set to wildcards (i.e., $*$). You are also given a non-negative integer r. The problem is to decide whether there exists an $m \times n$ matrix $\bar{M} \in \{0,1\}^{m \times n}$ (i.e., with **no** $*$**-entry**) such that

- For all $(i,j) \in \{1,2,\cdots,m\} \times \{1,2,\cdots,n\}$ with $M[i][j] \neq *$, $\bar{M}[i][j] = M[i][j]$, and
- The number of entries that are different between the last row and any other row in $\bar{M}$ is at most r. More formally, for every $i \in \{1,2,\cdots,m-1\}$,

$$\left|\{j \in \{1,2,\cdots,n\} \mid \bar{M}[i][j] \neq \bar{M}[m][j]\}\right| \leq r.$$

Your task is to show that the above problem is NP-complete. For that purpose, show the following.

(a) **[5 marks]** Show that the above problem is in NP.

(b) **[15 marks]** Prove that the above problem is NP-hard. (**Hints:** Try a reduction from the 3-SAT problem.)

Question 1 [20 marks]: Given a connected undirected graph G, a *spanning tree* is a connected subgraph that is a tree and includes all the vertices of G. Given any undirected graph, we all know how to compute a spanning tree (we can even compute a minimum spanning tree for weighted undirected graphs, but for those who can't recall that algorithm, that is fine, we do not need it for this question). Now, let us consider the following variant of the spanning tree problem.

Given a connected undirected graph G, the problem is to decide whether there exists a spanning tree with each node having degree at most five.

Your task is to show that the above problem is NP-complete. For that purpose, show the following.

(a) **[5 marks]** Show that the above problem is in NP.

(b) **[15 marks]** Prove that the above problem is NP-hard. (**Hints:** Try a reduction from the undirected Hamiltonian cycle problem.)

a) We can verify whether our graph contains a spanning tree with a list of edges. It takes polynomial time to verify that these edges connect the graph, that there are exactly n-1 edges, and no vertex is visited more than once. Since we can verify in polynomial time, this problem is NP.

b) We can prove that this problem is NP-Hard with, first, reduction from the Hamiltonian cycle to a Hamiltonian path:

Given a graph with a hamiltonian cycle, we omit the edge that takes us from the last vertex of the path, to the first vertex of the path

$G = (V, E) \in$ hamiltonian cycle. Once we remove the edge from (v_n, v_1), we have a hamiltonian path.

Hamitonian Cycle $\leq$ Hamiltonian Path

Now we reduce from Hamiltonian Path to spanning tree at most degree 5
Given an arbitiary graph G, let H be the graph obtained by adding a fan of 3 edges to every vertex of G. A spanning tree of H is almost-Hamiltonian if it has a maximum degree of 5. G contains a Hamiltonian path IFF H contains an almost-Hamiltonian spanning tree

Assume G has a hamiltonian path P. Let T be the spanning tree of H obtained by adding every fan edge in H to P. Every vertex v of H is either a leaf of T or a vertex of P. If $v \in P$, then $deg_P(v) \leq 2$, and therefore $deg_H(v) = deg_P(v) + 3 \leq 5$. This means H is an almost-Hamiltonian spanning tree.

Suppose H has an almost-Hamiltonian spanning tree T. The leaves of T are the vertrces of H w/ degree 1 which are also the vertices of H that aren't vertices of G. Let P be the subtree of T obtained by deleting every leaf of T. P is a spanning tree of G, and for every vertex $v \in P$, we have $deg_P(v) = deg_T(v) - 3 \leq 2$ so P is a Hamiltonian path in G.

This means:

$$\text{Hamiltonian Cycle} \leq \text{Hamiltonian Path} \leq \text{Spanning Tree of max degree 5}$$

Since the Hamiltonian Cycle problem is NP-Hard, so is this.

Question 2 [20 marks]: The presence of missing values in the data objects is ubiquitous in practice. Missing entries could be due to various types of faults during data collection, data transfer, or data cleaning phase and may occur arbitrarily. Dealing with a data set with missing values is one of the infamous challenges in data science. The problems of filling up missing entries are often cast as optimization tasks. One such problem is the following:

> You are given an $m \times n$ matrix M with each entry set to either 0 or 1 or a wildcard $*$. Furthermore, all the entries of the last (i.e., m^{th}) row of M are set to wildcards (i.e., $*$). You are also given a non-negative integer r. The problem is to decide whether there exists an $m \times n$ matrix $\bar{M} \in \{0,1\}^{m\times n}$ (i.e., with **no $*$-entry**) such that
>
> - For all $(i,j) \in \{1,2,\cdots,m\} \times \{1,2,\cdots,n\}$ with $M[i][j] \neq *$, $\bar{M}[i][j] = M[i][j]$, and
> - The number of entries that are different between the last row and any other row in $\bar{M}$ is at most r. More formally, for every $i \in \{1,2,\cdots,m-1\}$,
>
> $$\left|\{j \in \{1,2,\cdots,n\} \mid \bar{M}[i][j] \neq \bar{M}[m][j]\}\right| \leq r.$$

Your task is to show that the above problem is NP-complete. For that purpose, show the following.

(a) **[5 marks]** Show that the above problem is in NP.

(b) **[15 marks]** Prove that the above problem is NP-hard. (**Hints:** Try a reduction from the 3-SAT problem.)

a) We can verify in polynomial time $(n \cdot m)$ whether there is no missing data in the matrix represented by $*$ and in polynomial time $(2 \cdot n)$ whether difference between the last two rows is $\leq r$

b) We need to reduce the 3SAT problem to this problem to prove that it is NP-hard. Here is how to do the reduction

① Create a row in M for each clause in the 3-SAT formula and a column for each variable.

② Set each element in M equal to 1 if it's positive or 0 if negative. Set everything else to $*$. You're basically setting up the variables the same way they were set up in the original problem.

③ The 3SAT formula only works if the conditions are met for our original problem.

3SAT $\leq$ this problem

Since 3SAT is np-hard and this problem is atleast as hard as that, this problem is np-hard.