CSE 565: Algorithm Design and Analysis **Due 2022-11-04T23:59**

Problem Set 4

Instructor: Dr. Antonio Blanca *Release Date:* 2022-10-21
TA: Jeremy Huang

Notice: Type your answers using LaTeX and make sure to upload the answer file on Gradescope before the deadline. Recall that for any problem or part of a problem, you can use the "I'll take 20%" option. For more details and the instructions read the syllabus.

Problem 1. That's Bing

Suppose you have a procedure which runs in polynomial time and tells you whether or not a graph has a Hamiltonian path (given a graph $G = (V,E)$, whether there exists any $s,t \in V$ s.t. there exists a Hamiltonian-(s,t)-path in G). Show that you can use it to develop a polynomial-time algorithm for the Hamiltonian Path (which returns the actual path, if it exists).

Problem 2. 324

Assume that you are given a set of clauses where each of which is a disjunction of exactly 4 literals, and such that each variable occurs at most once in each clause. The goal is to find a satisfying assignment (if one exists). Prove that this problem is *NP*-complete.

Hint: Reduce 3SAT to this problem. Think about how to convert a clause from a 3SAT problem to an equivalent clause in this problem.

Problem 3. Brute Force

Show that for any problem $\Pi \in$ NP, there is an algorithm which solves Π in time $O(2^{p(n)})$, where n is the size of the input instance and $p(n)$ is a polynomial (which may depend on Π).

Problem 4. Mengsk

In an undirected graph $G = (V,E)$, we say $D \subseteq V$ is a dominating set if every $v \in V$ is either in D or adjacent to at least one member of D. In the DOMINATING SET problem, the input is a graph and a budget b, and the aim is to find a dominating set in the graph of size at most b, if one exists. Prove that this problem is NP-complete.

Hint: This problem is similar to the vertex cover problem.

Problem 5. Gethen

One experimental procedure for identifying a new DNA sequence repeatedly probes it to determine which k-mers (substrings of length k) it contains. Based on these, the full sequence must then be reconstructed. Let's now formulate this as a combinatorial problem. For any string x (the DNA sequence), let $\Gamma(x)$ denote the multiset of all of its k-mers. In particular, $\Gamma(x)$ contains exactly $|x| - k + 1$ elements. The reconstruction problem is now easy to state: given a multiset of k-length strings, find a string x such that $\Gamma(x)$ is exactly this multiset.

(1) Show that the reconstruction problem reduces to Hamiltonian PATH. (Hint: Construct a directed graph with one node for each k-mer, and with an edge from a to b if the last $k-1$ characters of a match the first $k-1$ characters of b.)

(2) Part 1 doesn't have to be bad news. Show that the same problem also reduces to EULER PATH. (Hint: This time, use one directed edge for each k-mer.)

Problem 6. Tricolor

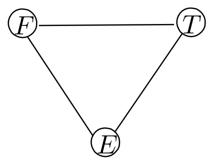
The k-coloring problem is the problem of deciding whether a graph $G = (V, E)$ is k-colorable – whether there exists an assignment c of colors $\{1, \ldots, k\}$ to the vertices V such that $c[u] \neq c[v]\ \forall (u, v) \in E$. That is, in a proper coloring, the endpoints of every edge are assigned different colors. Please follow the steps below to prove that 3COLOR, the 3-coloring problem, is NP-complete by reducing 3SAT to it.

Part (1)

Prove that 3COLOR is in NP.

Part (2)

Consider the following labeled 3-clique, denoted K_3:

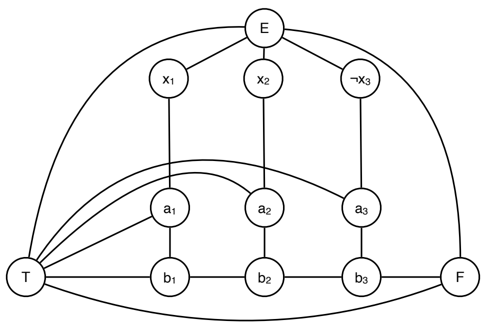


This graph has 3 nodes, and each node must be colored with a different color. We can use it to label the colors $\{1, 2, 3\}$ in a 3-coloring of K_3 as (T)rue, (F)alse, or (E)xcluded; that is, if vertex F is assigned color 2, then we associate color 2 with False.

Given a set of n boolean variables $x_1 \ldots x_n$, construct a graph G and a bijection between valid 3-colorings of G and all 2^n possible assignments of the boolean variables.

Part (3): Connecting One OR Gadget

Consider the following 12 node graph.



i. Argue that T, F, E all have different colors and that $x_1, x_2, \bar{x}_3$ must be colored with T or F.

ii. Prove that there is no valid 3-coloring of the graph where $\{x_1, x_2, \bar{x}_3\}$ all have the color F.

iii. Prove that for each other coloring of $\{x_1, x_2, \bar{x}_3\}$ (with T or F) there exists a compatible 3-coloring of the graph.

Part (4)

Given a set of n boolean variables $x_1 \ldots x_n$ and a set of m disjunctive clauses $c_1 \ldots c_m$ with three literals, describe a poly-time procedure that produces a graph which is 3-colorable if and only if all m clauses are satisfiable. Hint: connect multiple OR gadgets from Part (3) to the graph you made in Part (2).

Part (5)

Prove the correctness of your reduction and prove that it is a poly-time reduction. You should use the results from Part (3) in your proof.

Problem 7. Fixed Span

Determine which of the following problems are NP-complete and which are in P and explain why. In each problem you are given an undirected graph $G = (V, E)$, along with:
(a) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves includes the set L.
(b) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves is precisely the set L.
(c) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves is included in the set L.
(d) An integer k, and you must find a spanning tree with k or fewer leaves.

Problem 8. (I Can Get Some) Satisfaction

In the textbook, it has mentioned that 3SAT remains NP-complete even when restricted to formulas in which each literal appears at most twice.

(a.) Show that if each literal appears at most *once*, then the problem is solvable in polynomial time.

(b) Consider a special case of 3SAT in which all clauses have exactly three literals, and each variable appears at most three times. Show that this problem can be solved in polynomial time.

Problem 9. Stingy Set

(a.) STINGY_SAT is the following problem: given a set of clauses (each a disjunction of literals) and an integer k, find a satisfying assignment in which at most k variables are true, if such an assignment exists. Prove that STINGY SAT is NP-complete

(b.) Show that INDEPENDENT_SET remains NP-complete even in the special case when all nodes in the graph have degree at most 4.

Problem 10. Frozen Snake

A (2d) self-avoiding walk is a sequence of points $(p_1, \ldots, p_n)$ such that: 0. p_i are grid points – points with integer coordinates in $\mathbb{R}^2$, 1. no two points are equal, 2. p_i and p_{i+1} are exactly distance 1 apart, 3. $p_1 = (0, 0)$.

$A(n)$ is defined to be the number of unique self-avoiding walks of length n.

Describe a poly-space algorithm to compute $A(n)$ given n. Show that it works and that it has the correct space complexity. This will prove that the problem of computing $A(n)$ is in PSPACE.