Homework 7

CS 4104 (Spring 2017)

Assigned on April 21, 2021.
Submit PDF solutions on Canvas by the
11:59pm on **Friday**, April 30, 2021.

- For these problems, please describe the reduction as clearly as you can and make you sure you prove the correctness of the reduction in both directions, as we have discussed in class.

Problem 1 (20 points) Solve exercise 1 in Chapter 8 (page 505) of your textbook.

For each of the two questions below, decide whether the answer is (i) "Yes," (ii) "No," or (iii) "Unknown, because theit would resolve the question of whether $\mathcal{P} = \mathcal{NP}$." Give a brief explanation of your answer.

(a) Let's define the decision version of the Interval Scheduling Problem from Chapter 4 as follows: given a collection of intervals on a time-line, and a bound k, does the collection contain a subset of nonoverlapping intervals of size at least k?

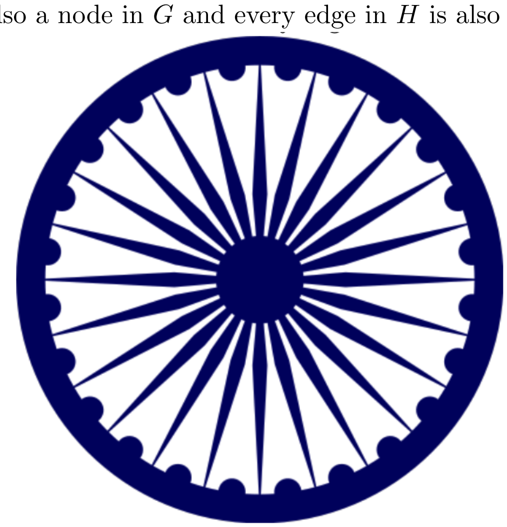
Question: Is it the case that Interval Scheduling $\leq_P$ Vertex Cover?

Solution: Yes. Since the Interval Scheduling problem can be solved in polynomial time and is therefore in NP, it can be reduced to the Vertex Cover problem since Vertex Cover is in $\mathcal{NP}$.

(b) Question: Is it the case that Independent Set $\leq_P$ Interval Scheduling

Solution: The answer is Unknown since Interval Scheduling can be solved in polynomial time. If Independent Set were polynomial-time reducible to Interval Scheduling, then that would imply that $\mathcal{P} = \mathcal{NP}$ since Independent Set is in $\mathcal{NP}$.

Problem 2 (30 points) The flag of a certain populous country contains a symbol called the "Ashoka Chakra" (see the image below). This symbol has a central hub and 24 spokes. Naturally, this reminds us of a graph with 25 nodes and 48 edges, of which 24 nodes are connected by a cycle, and the 25th node is connected to each of the other 24 nodes. A *generalised k-chakra* is a graph with $k+1$ nodes and $2k$ edges such that k nodes lie on a cycle and the $k+1$st node is connected to each of the other k nodes. Given an undirected graph G and an integer k, prove that the problem of determining if G contains a generalised k-chakra as a subgraph is $\mathcal{NP}$-Complete. (We say that a graph H is a *subgraph* of a graph G if every node in H is also a node in G and every edge in H is also an edge in G.)



I will get you started on the solution. Proving that k-chakra is in $\mathcal{NP}$ is easy. A certificate is just a candidate k-chakra. The certifier checks that this chakra contains $k+1$ nodes and $2k$ edges in the right configuration. The certificate takes $O(k)$ time to run.

Let us move on to proving that some $\mathcal{NP}$-Complete problem is reducible to the k-chakra problem. Suppose G has $n+1$ nodes. Let us consider the special case that $k = n$. An n-chakra looks suspiciously like a Hamiltonian cycle, except that the chakra has more edges. Therefore, let us reduce Hamiltonian cycle in undirected graphs (which we know to be $\mathcal{NP}$-Complete) to the k-chakra problem. Suppose H is an undirected graph that is an input to the Hamiltonian cycle problem. We want to convert it to a graph G that will be input to the k-chakra problem such that H contains a Hamiltonian cycle iff G contains a n-chakra. To complete the reduction, answer the following three questions:

(a) (7 points) Describe how you will convert an arbitrary undirected graph H that is input to the Hamiltonian cycle problem into an undirected graph G that is an input for the chakra problem.

Solution: To convert H into an undirected graph G, we just have to add the $n+1$ node to H and connect it to every other node in the graph to form the "spokes" of the n-chakra. Alternatively, if there already is a "center" node with spokes connecting to every other node, we just have to add this node to the perimeter and connect it to two other adjacent nodes (adjacent being connected by an edge) along the perimeter as well as to the central node.

(b) (8 points) If H contains a Hamiltonian cycle, prove that G contains a n-chakra.

Solution: If H contains a Hamiltonian cycle, we know that G must contain an n-chakra since the Hamiltonian cycle on H defines the perimeter and by our above construction, we know that by adding the "center" node and its spokes, we'll yield a the graph G with $n+1$ nodes and $2n$ edges.

(c) (15 points) If G contains a n-chakra, prove that H contains a Hamiltonian cycle. *Note:* there is a subtlety here that you have to be careful about.

Solution: There are two cases to consider

Case 1: If the $n+1$ node is connected to all other nodes in G, then it is the "center" of the chakra, so the rest of the nodes, and all of the edges not incident on this node are H.

Case 2: The "center" of the chakra is in H, and the $n+1$ node is part of the cycle in H. If this is the case, we can still find a Hamiltonian cycle of length n by traversing the other $n-1$ nodes, starting from and returning to the "center" node. We know that, despite it appearing as though we utilize the "spoke" edges of the n-chakra in G, all the edges used in the cycle must also exist in H since the $n+1$ node in G is *not* the "center" node.

Therefore, if G contains an n-chakra, H contains a Hamiltonian cycle.

Problem 3 (50 points = 25 + 25 points) Solve exercise 19 in Chapter 8 (pages 514–515) of your textbook. *Hint:* You can reduce 3-colouring to one problem (I am not saying which) and the other problem to network flow. Keep in mind that to reduce 3-colouring to one of these problems, you must take an undirected graph that you seek to colour with three colours and convert into a trucks-and-canister problem.

Suppose you're acting as a consultant for the port authority of a small Pacific Rim nation. They're currently doing a multi-billion-dollar business per year, and their revenue is constrained almost entirely by the rate at which they can unload ships that arrive in the port.

Handling hazardous materials adds additional complexity to what is, for them, an already complicated task. Suppose a convoy of ships arrives in the morning and delivers a total of n cannisters, each containing a different kind of hazardous material. Standing on the dock is a set of m trucks, each of which can hold up to k containers.

Here are two related problems, which arise from different types of constraints that might be placed on the handling of hazardous materials. For each of the two problems, give one of the following two answers:

- A polynomial-time algorithm to solve it; or

- A proof that it is NP-complete.

(a) For each cannister, there is a specified subset of the trucks in which it may be safely carried. Is there a way to load all n cannisters into the m trucks so that no truck is overloaded, and each container goes in a truck that is allowed to carry it?

Solution: We can answer this question with a polynomial-time algorithm via a reduction to the max flow problem.

We begin by building a flow network with nodes $v_c \in C$ for each cannister, nodes $v_t \in T$ for each truck, and edges (v_c, v_t) for each edge between a cannister and a truck which we add to a set E. Additionally, we add nodes s, t where s connects to every cannister node in C, and t connects to every truck node in T. We let $V = \Big\{C \cup T \cup \{s, t\}\Big\}$.

We then assign edge capacities of 1 to each edge between a cannister and a truck if and only if v_t is in the subset of trucks in which the cannister can be safely carried. We assign edge capacities of 1 for each edge between the source and a cannister, and edge capacity k for each edge between a truck and the sink node t to ensure that no trucks get overloaded.

From here, we simply run the Ford-Fulkerson algorithm on the flow network $G = (V, E)$. If an $s - t$ flow exists on this graph, then that means we can safely place all of the n cannisters onto the m trucks without violating the constraints of a flow network nor the constraints given in the problem. Otherwise, we know that there is no way to load the cannisters onto the trucks without violating one of the constraints.

Building such a flow network would take a polynomial amount of time, and the Ford-Fulkerson algorithm takes polynomial time, therefore this algorithm has a polynomial runtime complexity.

(b) In this different version of the problem, any cannister can be placed in any truck; however, there are certain pairs of cannisters that cannot be placed together in the same truck. (The chemicals they contain may react explosively if brought into contact.) Is there a way to load all n cannisters into the m trucks so that no truck is overloaded, and no two cannisters are placed in the same truck when they are not supposed to be?

Solution: We can reduce an instance of 3-Coloring to this problem to show that it is $\mathcal{NP}$-Complete.

First, we assert that the problem is at least in $\mathcal{NP}$ since we can certify that each truck carries a valid configuration of cannisters in polynomial time.

Next, to show that the problem is $\mathcal{NP}$-Complete, we demonstrate how to reduce an instance of 3-Coloring to the presented Trucks & Cannisters problem.

Assuming the input to the 3-Coloring problem is a graph G with $n = k$ nodes, we associate each node v_i with a cannister.

We assume that there are $m \geq 3$ colors to apply to the graph associated with the trucks which can hold k cannisters, where each truck is represented by a unique color.

We disallow two cannisters to be placed on the same truck if their corresponding nodes in G are connected. In other words, our 3-Coloring instance is conversely constructed such that no two nodes are connected if they cannot be placed in the same truck.

If there is a 3-Coloring on the graph G, then each color of a node corresponds to the truck that it can be safely placed within.

At this point we must show that the reduction is bi-directional, meaning that there is a solution to the given trucks and cannister problem if and only if there is a solution to our 3-Coloring problem.

3-Coloring $\rightarrow$ Trucks & Cannisters

If there is a solution to the 3-Coloring problem –a valid color assignment of each node v_i– then we can place all the cannisters of the same color inside of a truck and repeat for each color.

Trucks & Cannisters $\rightarrow$ 3-Coloring

If there is a way to place all the cannisters into the appropriate trucks, then there is also a solution to the 3-Coloring problem. We color the nodes on the graph according to the truck that each cannister was placed into.

Since we have shown that the 3-Coloring problem can be reduced to the Trucks & Cannisters problem and vice versa, then the presented problem is at least as difficult as the 3-Coloring problem and is therefore $\mathcal{NP}$-Complete.