Homework 7

CS 4104 (Fall 2021)

Assigned on November 29, 2021.
Submit PDF solutions on Canvas by the
11:59pm on December 7, 2021.

Instructions:

- The Honor Code applies to this homework with the following exception:
  - You can pair up with another student to solve the homework. Please form teams yourselves. Of course, you can ask the instructor for help if you cannot find a team-mate. You may choose to work alone.
  - You are allowed to discuss possible algorithms and bounce ideas with your team-mate. **Do not discuss proofs of correctness or running time in detail with your team-mate. You must write down your solution individually and indepedently. Do not send a written solution to your team-mate for any reason whatsoever.**
  - *In your solution, write down the name of the other member in your team. If you do not have a team-mate, please say so.*
  - Apart from your team-mate, you are not allowed to consult sources other than your textbook, the slides on the course web page, your own class notes, the TAs, and the instructor. In particular, do not use a search engine.
- Do not forget to typeset your solutions. *Every mathematical expression must be typeset as a mathematical expression, e.g., the square of n must appear as n^2 and not as "n^2".* You can use the LaTeX version of the homework problems to start entering your solutions.
- Do not make any assumptions not stated in the problem. If you do make any assumptions, state them clearly, and explain why the assumption does not decrease the generality of your solution.
- You must also provide a clear proof that your solution is correct (or a counter-example, where applicable). Type out all the statements you need to complete your proof. *You must convince us that you can write out the complete proof. You will lose points if you work out some details of the proof in your head but do not type them out in your solution.*
- If you are proposing an algorithm as the solution to a problem, keep the following in mind (the strategies are based on mistakes made by students over the years):
  - Describe your algorithms as clearly as possible. The style used in the book is fine, as long as your description is not ambiguous. Explain your algorithm in words. A step-wise description is fine. *However, if you submit detailed code or pseudo-code without an explanation, we will not grade your solutions.*
  - Do not describe your algorithms only for a specific example you may have worked out.
  - Make sure to state and prove the running time of your algorithm. You will only get partial credit if your analysis is not tight, i.e., if the bound you prove for your algorithm is not the best upper bound possible.
  - You will get partial credit if your algorithm is not the most efficient one that is possible to develop for the problem.
- In general for a graph problem, you may assume that the graph is stored in an adjacency list and that the input size is $m + n$, where n is the number of nodes and m is the number of edges in the graph. Therefore, a linear time graph algorithm will run in $O(m + n)$ time.

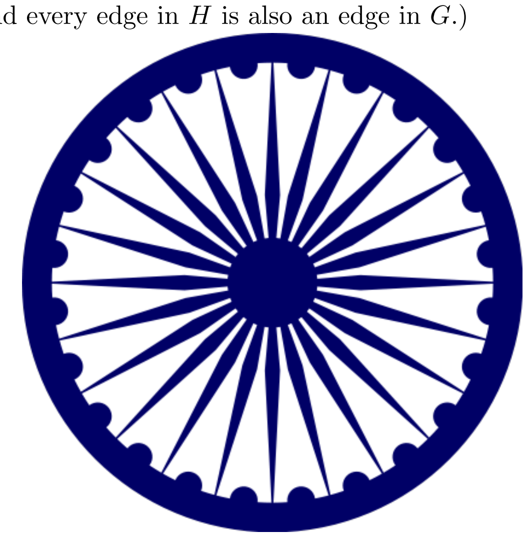
- For these problems, please describe the reduction as clearly as you can and make you sure you prove the correctness of the reduction in both directions, as we have discussed in class.

Problem 1 (20 points) Solve exercise 1 in Chapter 8 (page 505) of your textbook. Here is a sketch of the strategy you can follow for part (a). First reduce the decision version of the Interval Scheduling problem to the Independent Set problem.

1. Take an arbitrary input to Interval Scheduling, which is a set S of n intervals, each specified by a start time and finish time. How will you convert S to an undirected, unweighted graph G that is appropriate as an input to the Independent Set problem? Your approach should consist of statements that clearly map each interval to some part of G and each pair of non-overlapping intervals to some part of G.
2. What can you prove about S and G now? It should be a statement of the form "S contains at least k non-overlapping intervals if and only if G ..." You do not need to provide a proof of this statement.
3. Finally, how will you prove that Interval Scheduling is reducible to Vertex Cover? You need to state something we proved in class.

I will leave it to you to work out the solution for part (b).

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 *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 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 this problem is in $\mathcal{NP}$ is easy. A certificate is just a subgraph H of the input graph. The certifier checks that H contains $k+1$ nodes and $2k$ edges in the right configuration. The certifier runs in $O(k)$ time.

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.

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

(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. It is useful to think of G and H as distinct graphs with different sets of nodes and edges. How will you take the n-chakra in G and convert it back to a Hamiltonian cycle that involves only the nodes and edges of H?

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. So your solution should be clear about how you will convert nodes in the undirected graph (input to 3-colouring problem) into trucks or canisters, how will you represent edges in terms of trucks or canisters or canisters that should not be placed together, and what you will map the budget of 3 colours to? In the trucks-and-canister problem, there are no colours and the number three does not appear in the problem statement.