Complexity exercise set #5

for the tutorial on
May 19, 2022

Exercises marked with an asterisk ($*$) may be handed in for grading and can earn you a small bonus[1] on the exam, provided you submit your solutions via Brightspace in PDF before **15:15 on Monday May 30**.

Exercise 1 In this exercise, you get familiar with some notions introduced during the lecture.

1. Give a graph that does not have a clique of size 3.
2. Give a graph that does not have a 3-coloring.
3. Give a graph that has a 3-coloring and a clique of size 3.

Solution.

- We can, for example, take any graph of size less than 3, or any non-complete graph of size 3. Also any graph where all nodes have degree < 2, as a clique is a complete subgraph.
- Take, for example, K_n for $n > 3$.
- E.g., the complete graph of 3 nodes.

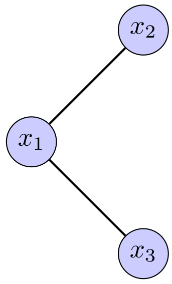
■

Exercise 2$*$ (50 points) In this exercise, we look at *independent sets* of graphs. Suppose, that we have a graph G. An **independent set** of G is a set X of vertices in G such that there is no edge from x to y if both x and y are members of X.

1. Give an example of a graph and two nonempty sets of vertices of which one is an independent set and the other is not.
2. Show that the following decision problem is **NP**-complete:
Given a graph G and an integer k, is there an independent set with at least k vertices?

Solution. For the first part, take e.g., the graph

[1] For more details, see `https://cs.ru.nl/~awesterb/teaching/2022/complexity.html`.



Then the set $\{x_2, x_3\}$ is an independent set, while $\{x_1, x_2\}$ (in fact, any set containing x_1) is not.

For the second part, we first show that the problem is in $\mathbf{NP}$. The certificate will be a set of vertices, which we check to see if it is independent and of size at least k. Checking the size can easily be done in linear time. To check for independence, we can take every pair of nodes in the set and check for an edge between them. The complexity of this step depends on the graph representation, but can certainly be done in polynomial time. Finally, such a certificate is polynomial in the size of the input as it will have size at most the number of nodes in the input graph. Thus, this problem is in $\mathbf{NP}$.

Next, we show that this problem is $\mathbf{NP}$-hard by showing that Clique can be reduced to it. Let G be a graph. Define a new graph $f(G)$ as follows:

- The nodes in $f(G)$ are the same as nodes in G
- There is an edge from x to y in $f(G)$ if and only if $x \neq y$ and there is no edge from x to y in G

Note: independent sets in $f(G)$ are the same as cliques in G, and cliques in $f(G)$ are the same independent sets in G. As such, G has a clique of size k if and only if $f(G)$ has an independent set of size k. The reduction is polynomial as we do not change the set of nodes and taking the complement of the set of edges can be done with complexity $O(n^2)$ where n is the number of nodes in G by considering all pairs of nodes in G and placing them in our new set of nodes if there is no edge between them in G. Thus, the problem is also $\mathbf{NP}$-hard, and so it is $\mathbf{NP}$-complete. ∎

Exercise 3$*$ (50 points) In this exercise, we look at the set cover problem. Suppose that we have a finite set $U \subset \mathbb{N}$ and a set S of subsets to U.

1. Take $U = \{1, \ldots, 10\}$ and take
$$S = \{\{2, 4, 7\}, \{2, 4, 8\}, \{1, 5, 7\}, \{8, 9\}, \{1, 3, 9\}, \{6, 10\}, \{1, 5, 8\}\}$$
Find a subset S' of S with size 4 such that the union of S' is U.
2. Show that the following problem is $\mathbf{NP}$-complete:
Given an integer k and sets U and S as above, is there a subset S' of S with k elements such that the union of the elements in S' is equal to U?

Solution. For the first:

$$\{2, 4, 7\} \cup \{1, 3, 9\} \cup \{6, 10\} \cup \{1, 5, 8\}$$

Next, we show that the given problem is in **NP**. The certificate will be some subset S' of S. To verify a candidate, we must check it has the correct size, is indeed a subset of S and that it covers U. The first two checks can easily be done in time linear in the size of S by stepping through the elements of S' to count them and check that they are contained in S. To check that S' covers U, we can consider each element of U and check that it is contained in a set in S'. This is linear in the size of U. Thus, the problem is in **NP**.

Now, we show that VertexCover reduces to this problem. Suppose, G is a graph and k is an integer. Write V for the set of vertices in G and without loss of generality, we assume that $V \subset \mathbb{N}$. Define:

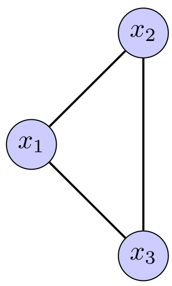
- $U = V$
- $S = \{\{w \in G \mid \text{there is an edge } e \text{ between } v \text{ and } w \text{ or } v = w\} \mid v \in G\}$

Note: the union of all elements in S is U. Also note: a vertex cover of G is essentially the same as a subset of S whose union is U. The reduction is polynomial as we only need to construct S, which can be done easily in polynomial time, for example, if G is represented by an adjacency list. Thus, the problem is also **NP**-hard, so also **NP**-complete. ■

Exercise 4 (more difficult) Let G be a connected graph whose set of vertices is V and whose set of edges is E. A set $X \subseteq V$ is called **dominating** if for every vertex $v \in V$ we either have $v \in X$ or there is an $x \in X$ and an edge between v and x.

1. Give an example of a dominating set that is not a vertex cover
2. Show that the following problem is **NP**-complete:
   Given a graph G and an integer k, is there a dominating set of size at most k?

Solution. Consider the following graph



The set $\{x_1\}$ is a dominating set but not a vertex cover as the edge (x_2, x_3) has no endpoints in the set.

We now show that the given problem is in **NP**. A certificate will be a subset X of the nodes of G. We need to verify that it is of the correct size, and that it is dominating. Checking size is again done easily in linear time. To check that it dominates G, we take each $v \in V$ and check if it is contained in X. If it is, we continue, if it is not, we check each node adjacent to v to see if it is in X. The complexity of this check is $O(n^2)$ where n is the number of nodes in G. We see that the problem is indeed in **NP**.

Next, we show that VertexCover reduces to this problem. Let G be a graph with vertices V and edges E, and let k be an integer. Define a graph G'

- The set of vertices is $V \cup E$
- For every edge e between v and w, we have edges between v and e, between w and e, and between v and w

Suppose, we have a dominating set D for G' of size k. Consider the following set:

$$C' = (V \cap D) \cup \{v \mid \text{ for } e \in E \text{ between } v \text{ and } w \text{ with } e \in D\ \}$$

Note: that we pick one side for every edge in D, and for that reason C' has k elements.

Let $e \in E$ be an edge between v and w. Then there are two cases:

- $e \in D$. In that case, we chose one of the sides of e which is in C'
- $e \notin D$. In that case, there is an edge from e to some element in D. By the way we defined the graph, this must be either v or w if e is an edge between v and w. Hence, either v or w is in V, so either v or w is in C'.

Hence C' is a vertex cover.

Suppose that we have a vertex cover C for G of size k. Note: C is a dominating set for G'. Let v be a vertex in G'. There are two cases:

- This vertex comes from an edge e between v and w. Since C is a vertex cover, either v or w is in C. Hence, we have the desired edge.
- Suppose, this vertex comes from a vertex v. Since G is connected, there is an edge between v and some other vertex w. Because C is a vertex cover, either v or w is in C, and in this case, we thus have the desired edge.

Thus, the problem is also **NP**-hard, so also **NP**-complete by the above.

■