# COMS20010 — Problem sheet 9

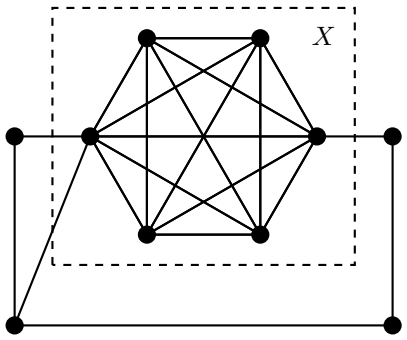
You don't have to finish the problem sheets before the class — focus on understanding the material the problem sheet is based on. If working on the problem sheet is the best way of doing that, for you, then that's what you should do, but don't be afraid to spend the time going back over the quiz and videos instead. (Then come back to the sheet when you need to revise for the exam!) I'll make solutions available shortly after the problem class. As with the Blackboard quizzes, question difficulty is denoted as follows:

- ⋆ You'll need to understand facts from the lecture notes.
- ⋆⋆ You'll need to understand and apply facts and methods from the lecture notes in unfamiliar situations.
- ⋆⋆⋆ You'll need to understand and apply facts and methods from the lecture notes and also move a bit beyond them, e.g. by seeing how to modify an algorithm.
- ⋆⋆⋆⋆ You'll need to understand and apply facts and methods from the lecture notes in a way that requires significant creativity. You should expect to spend at least 5–10 minutes thinking about the question before you see how to answer it, and maybe far longer. At most 10% of marks in the exam will be from questions set at this level.
- ⋆⋆⋆⋆⋆ These questions are harder than anything that will appear on the exam, and are intended for the strongest students to test themselves. It's worth trying them if you're interested in doing an algorithms-based project next year — whether you manage them or not, if you enjoy thinking about them then it would be a good fit.

If you think there is an error in the sheet, please post to the unit Team — if you're right then it's much better for everyone to know about the problem, and if you're wrong then there are probably other people confused about the same thing.

This problem sheet covers week 10, focusing on complexity theory.

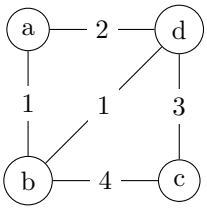
1. [⋆⋆] A *clique* in a graph $G = (V, E)$ is a set of vertices $X \subseteq V$ in which every two vertices are joined by an edge, as in the picture below.



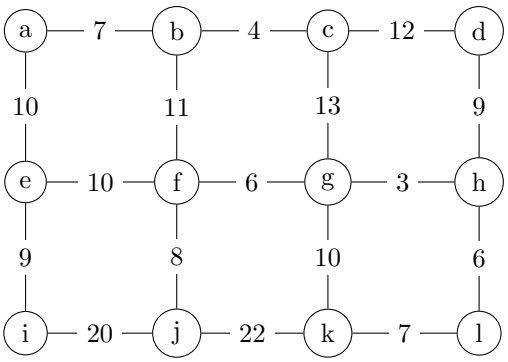

   Prove that the following problem is NP-complete under Karp reductions: given a graph $G = (V, E)$ and an integer $k$, does $G$ contain a clique of size at least $k$? (**Hint:** To show hardness, try reducing from IS.)

2. Suppose we are given a weighted directed graph $G$ and an integer $k$, and we want to know whether or not $G$ contains a Hamilton cycle of length at most $k$ — this is called the Travelling Salesman Problem (TSP). Recall that a Hamilton cycle is a cycle which starts and ends at the same vertex, and visits every vertex exactly once.

(a) [⋆⋆] Does this graph have a Hamilton cycle of length less than 5?




(b) [⋆⋆] Consider the following graph.




Now consider the Hamilton cycle *abcdhlkgfjiea*. Does this graph have a Hamilton cycle of length less than 130? In general, if we're given a Hamilton cycle for $G$, can we determine its length in time polynomial in the input size?

(c) [⋆⋆] What does this tell us about TSP in terms of P and NP?

(d) [⋆⋆] Prove that TSP is NP-complete. (**Hint:** Check out question 11b).)

3. [⋆⋆] You are attempting to form a committee of people to represent Computer Science in an upcoming faculty restructure. A regrettably large number of stakeholders have ideas about what this committee should look like: Denoting the stakeholders by $S_1, \ldots, S_n$, each stakeholder $S_i$ has one list $S_i^+$ of people they would like to be on the committee, and another list $S_i^-$ of people they would like not to be on the committee. (Either list may be empty.) As in real life, the committee can be arbitrarily large.

   You quickly realise that it is impossible to satisfy everyone's preferences at once, so you decide to try for something easier: you are trying to grant every person at least one of their requests. Thus for every stakeholder $S_i$, either you have added a person in $S_i^+$ to the committee, or you have *not* added a person in $S_i^-$ to the committee, or both. For example, if everyone requested that John Lapinskas be added to the committee, then any committee containing John would be a valid committee. Sketch a proof that even so, deciding whether or not a valid committee exists given $S_1^+, \ldots, S_n^+$ and $S_1^-, \ldots, S_n^-$ is an NP-complete problem.

4. In this question we will show that if SAT Karp-reduces to $\overline{\text{SAT}}$, then NP = Co-NP.

   (a) [⋆⋆] Suppose we have an oracle for $\overline{\text{SAT}}$ (the complement of SAT). Show how we can use this to construct a Cook reduction from SAT to $\overline{\text{SAT}}$.

   (b) [⋆⋆] Show that if there is a Karp reduction from SAT to $\overline{\text{SAT}}$, then SAT $\in$ Co-NP and $\overline{\text{SAT}} \in$ NP.

   (c) [⋆⋆] Show that if there is a Karp reduction from SAT to $\overline{\text{SAT}}$, then NP = Co-NP. (**Hint:** Show that NP $\subseteq$ Co-NP and Co-NP $\subseteq$ NP.)

5. (a) [⋆⋆] Consider a set $U = \{1, \ldots, n\}$, and subsets $S_1, \ldots, S_m \subseteq U$, and a number $k \in \mathbb{Z}$. The Set Cover problem asks if there exists a collection $C$ of at most $k$ sets in $U$ such that their union covers all of $U$. Show that VertexCover $\leq_K$ SetCover.

   (b) [⋆⋆] The Set Packing problem is similar to the Set Cover problem, except this time we are asking if there is a subset of at *least* $k$ sets in $U$ such that no two sets in $C$ intersect. Show that IndependentSet $\leq_K$ SetPacking.

(c) [★★] Now show that VertexCover $\leq_C$ SetCover and IndependentSet $\leq_C$ SetPacking.

6. In this question, we work with a variant of SAT in which variables cannot be negated. Given literals $a$, $b$ and $c$, which need not be distinct, an *even clause* EVEN$(x, y, z)$ evaluates to `True` if and only if either zero or two of $x$, $y$ and $z$ evaluate to `True`. A *width-3 positive OR clause* is an OR clause of three variables (i.e. **un-negated** literals). A *positive even formula* is a conjunction of even clauses and width-3 positive OR clauses. For example,

$$\text{EVEN}(a, \neg b, c) \wedge \text{EVEN}(a, a, d) \wedge (a \vee b \vee e) \wedge \text{EVEN}(c, d, e).$$

is a positive even formula, but $(\neg a \vee b)$ is not due to both the negated variable and the fact that the clause only contains two variables. The decision problem POS-EVEN-SAT asks whether a positive even formula (given as the input) is satisfiable, in which case the desired output is `Yes`.

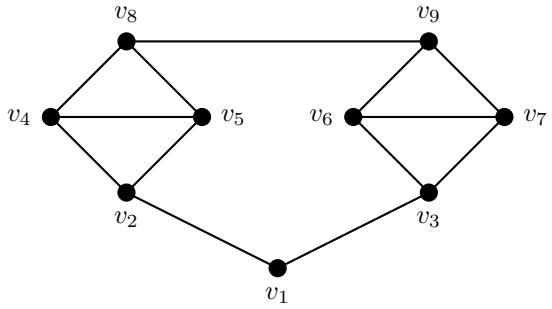
(a) [★★] Give a Karp reduction from POS-EVEN-SAT to SAT and briefly explain why it works.

(b) [★★★] Give a Karp reduction from 3-SAT to POS-EVEN-SAT and briefly explain why it works.

7. [★★★] Suppose you're working at a hardware store which is trying to put a market research group together. They serve a lot of markets, from woodworking to metalworking to painting, so they'd like their group to be diverse. More specifically, they'd like no two people in the group to have bought the same product from them within the last year. Subject to this, they'd like the group to be as large as possible. They've hired you to develop an algorithm to do this for them. They give you a list of products, a list of customers, and their sales data for the past year in the format of a two-dimensional array $L$, where $L[i][j]$ is the number of copies of product $j$ bought by customer $i$. Their desired output is a diverse focus group which is as large as possible. Show that this problem is NP-hard under Cook reductions. (Note: There are **many** more questions like this in Chapter 8 of Kleinberg and Tardós.)

8. (a) [★★★] Prove that IS is NP-complete under Karp reductions even when the input graph is required to have maximum degree at most 3. **Hint:** Try to adapt the NP-completeness proof for IS given in lectures.

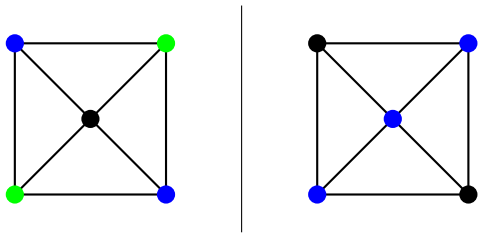
(b) [★★★★] Prove that IS is NP-complete under Karp reductions even when the input graph is required to be cubic — that is, when every vertex has degree *exactly* 3. **Hint:** You may find the following graph useful as a gadget:




(c) [★★] Is the problem of finding a maximum independent set NP-hard when the input graph has maximum degree 2?

9. [★★★] In lectures, we proved that integer linear programming was NP-hard by a long chain of reductions. Give an alternative proof via a direct Cook reduction from 3-SAT to integer linear programming. (**Hint:** The idea is similar to the reduction from vertex cover.)

10. For all positive integers $k$, a *$k$-colouring* of a graph $G = (V, E)$ is a map $c\colon V \to C$, where $C$ is a set of *colours* with $|C| = k$. We say $c$ is *proper* if $c(u) \neq c(v)$ whenever $\{u, v\} \in E$, i.e. no two adjacent vertices receive the same colour. Two colourings of the same graph, with $C \subseteq \{\text{green}, \text{blue}, \text{black}\}$, are shown below. The 3-colouring on the left is proper, but the 2-colouring on the right is not (because the blue center vertex is adjacent to blue vertices at the corners).

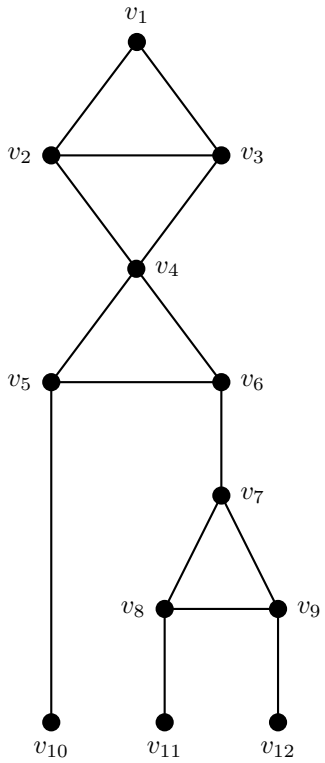


This question is concerned with the following **family** of decision problems. Let $k$ be a positive integer. Then the $k$-colourability problem asks: given a graph $G$, does $G$ have a proper $k$-colouring? Note that $k$ is not part of the input, but part of the problem statement.

(a) [$\star\star$] Give a polynomial-time algorithm for the 2-colourability problem.

(b) [$\star\star\star$] Next, for all $k > 3$, give a Karp reduction from the 3-colourability problem to the $k$-colourability problem.

(c) [$\star\star$] In fact, 3-colourability is NP-hard under Karp reductions. One way of proving it is to reduce directly from 3-SAT. A crucial gadget in the standard reduction is as follows:




Note that in any proper 3-colouring, the vertices $v_1$, $v_2$ and $v_3$ must receive different colours. Write $T$ for the colour of $v_1$, $F$ for the colour of $v_2$, and $B$ for the colour of $v_3$. Prove that the gadget has the following properties:

- In any proper 3-colouring of the gadget, the vertices $v_{10}$, $v_{11}$ and $v_{12}$ cannot all be coloured $F$.
- Any other colouring of $v_{10}$, $v_{11}$ and $v_{12}$ using only colours $T$ or $F$ can be extended to a proper colouring of the entire gadget.

**Hint:** There are two ways of proving the second part — a long-but-simple way and a short-but-clever way.

(d) [$\star\star\star\star$] Using the gadget from the previous part, or otherwise, prove that 3-colourability is NP-complete under Karp reductions.

11. (a) [★★★★] Prove that the problem of deciding whether or not a **directed** graph contains a (directed) Hamilton cycle is NP-complete under Karp reductions. **Hint:** One valid approach involves replacing variables by path gadgets with edges in both directions, where travelling one direction on the path corresponds to an assignment of true and travelling the other direction corresponds to an assignment of false, and replacing clauses with individual vertices.

    (b) [★★★] Prove that the problem of deciding whether or not an **undirected** graph contains a Hamilton cycle is NP-complete under Karp reductions. **Hint:** One valid approach uses small vertex gadgets.