Compsci 330 Design and Analysis of Algorithms
Assignment 9, Spring 2024 Duke University

TODO: Add your name(s) here

Due Date: Thursday, April 4, 2024

How to Do Homework. We recommend the following three step process for homework to help you learn and prepare for exams.

1. Give yourself 15-20 minutes per problem to try to solve on your own, without help or external materials, as if you were taking an exam. Try to brainstorm and sketch the algorithm for applied problems. Don't try to type anything yet.
2. After a break, review your answers. Lookup resources or get help (from peers, office hours, Ed discussion, etc.) about problems you weren't sure about.
3. Rework the problems, fill in the details, and typeset your final solutions.

Typesetting and Submission. Your solutions should be typed and submitted as a single pdf on Gradescope. Handwritten solutions or pdf files that cannot be opened will not be graded. LaTeX[1] is preferred but not required. You must mark the locations of your solutions to individual problems on Gradescope as explained in the documentation. Any applied problems will request that you submit code separately on Gradescope to be autograded.

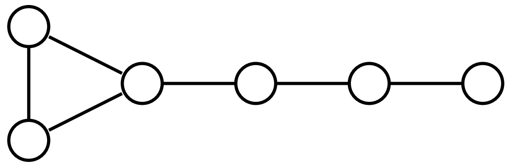
Writing Expectations. If you are asked to provide an algorithm, you should clearly and unambiguously define every step of the procedure as a combination of precise sentences in plain English or pseudocode. If you are asked to explain your algorithm, its runtime complexity, or argue for its correctness, your written answers should be clear, concise, and should show your work. Do not skip details but do not write paragraphs where a sentence suffices.

Collaboration and Internet. If you wish, you can work with a single partner (that is, in groups of 2), in which case you should submit a single solution as a group on gradescope. You can use the internet, but looking up solutions or using large language models is unlikely to help you prepare for exams. See the course policies webpage for more details.

Grading. Theory problems will be graded by TAs on an S/U scale (for each sub-problem). Applied problems typically have a separate autograder where you can see your score. The lowest scoring problem is dropped. See the course assignments webpage for more details.

[1] If you are new to LaTeX, you can download it for free at latex-project.org or you can use the popular and free (for a personal account) cloud-editor overleaf.com. We also recommend overleaf.com/learn for tutorials and reference.

Problem 1 (Kite). A *kite* is a graph on an even number of vertices, say $2n$, in which n of the vertices form a clique and the remaining n vertices are connected in a "tail" that consists of a path joined to one of the vertices of the clique. An example of a kite with 6 vertices is diagrammed below.



Given a graph and a goal g, the KITE problem asks whether there is a subgraph which is a kite and which contains $2g$ nodes. Prove that KITE is NP-complete. Your reduction should be from one of the following problems: INDEPENDENT SET, VERTEX COVER, CLIQUE, HAMILTONIAN CYCLE, or HAMILTONIAN PATH.

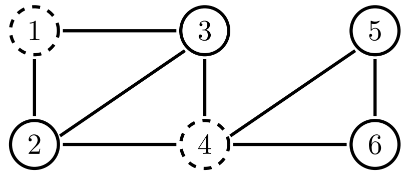
Problem 2 (Contracts). You work for a company that takes consulting jobs but has more contracts on offer than they can manage. Specifically, you have n contract offers numbered $1, 2, \ldots, n$, each of which earns a value of v_i dollars and requires a work time of w_i person-hours. Given these, a financial goal g, and a time bound t, the CONTRACTS problem asks whether there is a set of contracts C with total work time at most t (that is, $\sum_{i \in C} w_i \leq t$) and value at least g (that is, $\sum_{i \in C} v_i \geq g$).

For example, if the following three contracts are available, then given $t = 10$ and $g = 600$, the answer would be TRUE, because choosing contracts 1 and 2 would suffice. However, if $t = 10$ but $g = 700$, the answer would be FALSE because no set of contracts with total work time at most 10 gets total value at least 700.

Contract (i)	Value v_i	Work w_i
1	200	4
2	400	5
3	500	9

Prove that this problem is NP-complete. Your reduction should be from one of the following problems: 3SAT, SUBSET SUM, or PARTITION.

Problem 3 (Cycle Interrupting). Let $G = (V, E)$ be a connected undirected graph. Say that a subset of vertices $I \subseteq V$ is called *cycle interrupting* in G if every cycle in G contains at least one vertex of I. For example, in the graph below, the dashed vertices 1 and 4 constitute a cycle interrupting set of size 2.



The CYCLE INTERRUPTING problem asks, given an undirected graph G and an integer k as input, whether G contains a cycle interrupting set of size at most k. Prove that this problem is NP-complete. Your reduction should be from one of the following problems: INDEPENDENT SET, VERTEX COVER, CLIQUE, HAMILTONIAN CYCLE, or HAMILTONIAN PATH.

Problem 4 (Binary Constraint Programming). Given integer constraint coefficients a_{ij} for every i from 1 to m and j from 1 to n, as well as m integer constraint terms $b_1, b_2, \ldots, b_m$, the BINARY CONSTRAINT PROGRAMMING (or BCP for short) problem asks whether there exists an assignment of 0 or 1 to each of n variables $x_1, x_2, \ldots, x_n$ such that $\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i$ for every i from 1 to m.

For example, suppose $n = 3$, $m = 2$, and the input is

$$\begin{bmatrix} a_{11} = 5 & a_{12} = -2 & a_{13} = 0 \\ a_{21} = -3 & a_{22} = 3 & a_{23} = 4 \end{bmatrix} \qquad \begin{bmatrix} b_1 = 4 \\ b_1 = 0 \end{bmatrix}$$

Then the answer would be TRUE, because setting $x_1 = 1$, $x_2 = 1$, and $x_3 = 0$ satisfies both constraints:

- $a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = (5)(1) + (-2)(1) + (0)(0) = 3 \leq 4$, and
- $a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 = (-3)(1) + (3)(1) + (4)(0) = 0 \leq 0$, and

Prove that BLOPT is NP-complete. Your reduction should be from one of the following problems: 3SAT, INDEPENDENT SET, SUBSET SUM, or PARTITION.