

Homework 5 (125 points)

Out: Monday, April 18, 2022

Due: 11:59pm, Monday, May 2, 2022

Homework Instructions.

1. For all algorithms that you are asked to “give” or “design”, you should
 - Describe your algorithm clearly in English.
 - Give pseudocode.
 - Argue correctness; you may provide a formal proof or a convincing argument.
 - Provide, with an explanation, the best (smallest) upper bound that you can for the running time. All bounds should be **worst-case** bounds, unless explicitly stated otherwise.

You are also encouraged to analyze the space required by your algorithm. We will not remove marks if you don’t, unless the problem explicitly asks you to analyze space complexity.

2. If you give a dynamic programming algorithm, you should follow the instructions in HW2.
3. If you give a **reduction**, you should do so as we did in class, that is
 - (a) Give the inputs to the two problems.
 - (b) Describe in English the reduction transformation and argue that it requires polynomial time. (You do not need to give pseudocode.)
 - (c) Prove carefully equivalence of the original and the reduced instances.
4. **You should not use any external resources for this homework. You should work on these problems entirely on your own and avoid collaborating with your classmates**, unless you have thought through the problems on your own for a long time and are unable to make any further progress. Failure to follow this instruction might have a negative impact on your performance in the second exam and interviews.
5. You should submit this assignment as a **pdf** file to Gradescope. Other file formats will not be graded, and will automatically receive a score of 0.
6. I recommend you type your solutions using LaTeX. For every assignment, you will earn 5 extra credit points if you type your solutions using LaTeX or other software that prints equations and algorithms neatly. If you do not type your solutions, make sure that your handwriting is very clear and that your scan is high quality.
7. You should write up the solutions **entirely on your own**. Collaboration is limited to discussion of ideas only. You should adhere to the department’s academic honesty policy (see the course syllabus). Similarity between your solutions and solutions of your classmates or solutions posted online will result in receiving a 0 in this assignment, and possibly further disciplinary actions. There will be no exception to this policy and it may be applied retro-actively if we have reasons to re-evaluate this homework.

Homework Problems

1. (30 points) You are asked to assist in the following crisis event.

Due to large scale flooding, there is a set of n injured people distributed across a region that need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now). However you cannot overload any single hospital; instead, every hospital must receive at most $\lceil n/k \rceil$ people.

Give an efficient algorithm for this problem.

2. (25 points) In the **min-cost flow** problem, the input is a *flow network with supplies* where each edge $(i, j) \in E$ also has a *cost* a_{ij} , that is, sending one unit of flow on edge (i, j) costs a_{ij} .

Given a flow network with supplies and costs, the goal is to find a feasible flow $f : E \rightarrow R^+$, that is, a flow satisfying edge capacity and node supply constraints, that *minimizes* the total cost of the flow.

Show that the max flow problem can be formulated as a min-cost flow problem.

3. (25 points) Suppose you had a polynomial-time algorithm A for the decision version **VC(D)** of the minimum vertex cover problem. That is, on input a graph $G = (V, E)$ and an integer k , A answers **yes** if and only if G has a vertex cover of size at most k , and A runs in worst-case polynomial time.

Design a polynomial-time algorithm that takes as input a graph G and an integer k , and returns a vertex cover of size at most k , if one exists in G , or **no** otherwise.

4. (20 points) In the **MAX SAT** problem, the input is a **SAT** formula ϕ with m clauses over n variables and the output is a truth assignment that maximizes the number of satisfied clauses, as well as that number.

State the decision version of **Max SAT** and show that it is \mathcal{NP} -complete.

5. (25 points) A *clique* in an undirected graph $G = (V, E)$ is a subset S of vertices such that *all* possible edges between the vertices in S appear in E .

Max Clique is the following problem: On input a graph $G = (V, E)$, find a clique of maximum size.

Computing the maximum clique in a network (or the number of cliques of at least a certain size) is useful in analyzing social networks, where cliques corresponds to groups of people who all know each other.

State the decision version of **Max Clique** and show that it is \mathcal{NP} -complete.

RECOMMENDED exercise: do NOT return, it will not be graded.

1. Suppose you are given n cities and a set of non-negative distances d_{ij} between pairs of cities.
 - (a) Give an $O(n^2 2^n)$ dynamic programming algorithm to solve this instance of **TSP**; that is, compute the cost of the optimal tour and output the actual optimal tour.
 - (b) What are the space requirements of your algorithm?

Hint: Let $V = \{1, \dots, n\}$ be the set of cities. Consider progressively larger subsets of cities; for every subset S of cities including city 1 and at least one other city, compute the shortest path that starts at city 1, visits all cities in S and ends up in city j , for every $j \in S$.