CS256: Algorithm Design and Analysis **Spring 2021**

Assignment 7 (due 04/28/2021)

Instructor: Shikha Singh *Solution template:* Overleaf

Note. This assignment would typically be due in a week after it is released, but in order to give students some respite around Health Days (April 21 and 22), it is due the week after. However, it is a tad longer than a typical assignment. To break up the workload, I *strongly encourage* students to attempt Questions 1-3 during the week of April 14-April 21, and Questions 4-6 during the week of April 21-28.

Network Flow Applications

Flow reductions. In the first two question, you must solve the given problem by reducing it to a flow problem. Structure your solution to include the following pieces:

- State the reduction: map an arbitrary instance of the given problem to a special instance of a flow problem
- Prove the correctness of the reduction: show that solution of the given problem has a 1-1 correspondence to the solution of the max-flow problem on the created flow network (this is an *if and only if* proof, that is, you must prove both directions of the mapping)
- Analyze the running time of the overall solution, that is, running time of reduction + time taken by the flow algorithm (which often dominates)

Problem 1. (KT 7.7) Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible base stations. We'll suppose there are n clients, with the position of each client specified by its (x, y) coordinates in the plane. There are k base stations; the position of each of these is specified by (x, y) coordinates as well.

For each client, we wish to connect it to exactly one of the base stations. Our choice of connections is constrained in the following ways.

- There is a range parameter r—a client can only be connected to a base station that is within distance r.
- There is a load parameter L—no more than L clients can be connected to any single base station.

Your goal is to design a polynomial time algorithm for the following problem. Given the positions of a set of clients and a set of base stations, as well as the range and load parameters, decide whether every client can be connected simultaneously to a base station, subject to the range and load conditions in the previous paragraph.

Problem 2. (From Dave Mount's Algorithms Class) The computer science department at a major university has a tutoring program. There are m tutors, $\{t_1, \ldots, t_m\}$ and n students who have requested the tutoring service $\{s_1, \ldots, s_n\}$. Each tutor t_i has a set T_i of topics that they know, and each student s_j has a set of topics S_j that they want help with. We say that tutor t_i is suitable to work with student s_j if $S_j \subseteq T_i$ (That is, the tutor t_i knows all the topics of interest to student s_j.) Finally, each tutor t_i has a range $[a_i, b_i]$, indicating that t_i would like to work with at least a_i students (so their time is well-utilized) and at most b_i students (so they are not overburdened).

Given a list of students, a list of tutors, the ranges $[a_i, b_i]$ for the tutors, and a list of suitable tutors for each student, describe an efficient algorithm that determines whether it is possible to generate a pairing of tutors to students such that:

- Each student is paired with exactly one tutor.
- Each tutor t_i is paired with at least a_i and at most b_i students.
- Each student is paired only with a suitable tutor.

Classes P and NP

Problem 3. Suppose an extra-terrestrial being gave you a magic blackbox that can solve HAMILTONIAN-CYCLE, which is an NP-complete problem, with a worst-case running time of $O(n^8)$, where n denotes the number of vertices in the graph.

Assuming that our magic box actually works, which of the following can we conclude? Give a brief justification with your response.

(a) All problems in NP are solvable in polynomial time.
(b) All NP-hard problems are solvable in polynomial time.
(c) All NP-complete problems are solvable in $O(n^8)$ time.

NP-Completeness

A Note of NP-completeness proofs. The next three questions will ask you to prove that a given problem X is NP Complete. A complete solution would include the following:

- Problem X is in NP: you don't have to give a "tight bound" on the verification complexity just briefly reason why it can be achieved in time polynomial in the input size.

- State a *known* NP hard problem Y from class that you will use to prove X is also NP hard

- Show that $Y \leq_p X$. Remember to:

  - Prove that the reduction is correct by arguing both the "if" and "only if" directions

  - Argue that your reduction is polynomial time (again, you are don't have to give an efficient bound, just that is is polynomial time, so this piece should 1-2 sentences)

Students often ask "Which problems can I use as the *known* NP-hard problem?" You can use any of the problems whose NP-hardness was established in class, or Chapter 8 of Kleinberg Tardos or Chapter 12 of Erickson. Here is a (not necessarily complete) list:

(a) Independent Set
(b) Vertex Cover
(c) Set Cover
(d) Clique
(e) Circuit-SAT/SAT/3-SAT
(f) Hamiltonian Cycle/Path
(g) Graph 3-color
(h) Subset-Sum

Problem 4. (KT 8.5) Consider a set $A = \{a_1, \ldots, a_n\}$ and a collection $B_1, B_2, \ldots, B_m$ of subsets of A (i.e., $B_i \subseteq A$ for each i). We say that $H \subseteq A$ is a hitting set for the collection $B_1, \ldots, B_m$ if H contains at least one element from each B_i—that is, if $H \cap B_i$ is not empty for each i (so H "hits" all the sets B_i).

We now define the *Hitting Set Problem* as follows: given $A = \{a_1, \ldots, a_n\}$, a collection $B_1, B_2, \ldots, B_m$ of subsets of A, and a number k, is there a hitting set $H \subseteq A$ for $B_1, B_2, \ldots B_m$ such that size of H is at most k. Prove that Hitting Set is NP complete.

Problem 5. Suppose you are in charge of forming a student committee at Williams. Given a set of n students and a compatibility function that maps every student to a subset of students they are compatible with, you must choose k students to be in the committee such that each student in the committee is compatible with everyone else.

(a) Show that the problem of determining whether it is possible to form such a committee is NP-complete.

(b) Suppose you were forced to pick a given student as the president of your committee. That is, you must always include this person (and each person in the committee should still be compatible with the president and everyone else). Show that this new version is still NP-complete, by modifying your reduction from part (a).

Note. You need to use a polynomial-time reduction as defined in class: you need to map "yes"-instances to "yes"-instances, and "no"-instances to "no"-instances. Any answers of the type: "choose each vertex one at a time as president and call the oracle n times" are not correct polynomial time reductions.

Problem 6. (KT 8.28) The following is a version of the Independent Set problem. You are given an undirected connected graph $G = (V, E)$ and an integer k. For this problem, we will call a set $I \subseteq V$ *strongly independent* if for any two nodes $v, u \in I$, the edge (v, u) does not belong to E, and there is also no path of 2 edges from u to v, i.e., there is no node w such that both $(u, w) \in E$ and $(w, v) \in E$. The Strongly Independent Set problem is to decide whether G has a strongly independent set of size at least k. Prove that the Strongly Independent Set problem is NP-complete.

Problem 7. (**Extra Credit: 5 points**) Define the $\mathsf{ODD-SUBSET-SUM-RANGE}$ problem as follows. Given n odd numbers $s_1, \ldots s_n$, and two target integers T_1 and T_2, determine if there exists a subset of numbers that adds up to a value between T_1 and T_2.[1] Prove that $\mathsf{ODD-SUBSET-SUM-RANGE}$ is NP-hard.

[1] "Between" is inclusive, so a subset adding up to exactly T_1 or T_2 indicates a yes instance.