Homework 2 (145 points)

Out: Friday, February 5, 2021
Due: 11:59pm, Friday, February 19, 2021

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.
   - Give the best upper bound that you can for the running time.

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

2. You should submit this assignment as a **pdf** file on Gradescope. Other file formats will not be graded, and will automatically receive a score of 0.

3. 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 hand-writing is very clear and that your scan is high quality.

4. 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. (25 points) There is a natural intuition that, if two nodes are "far apart" in a communication network, they may become disconnected more easily. One way of formalizing this intuition is the following. You are given an undirected graph $G = (V, E)$ with n vertices and m edges that contains two nodes s and t such that the distance between s and t is strictly greater than $n/2$. Give an efficient algorithm that returns a node v different from both s and t such that deleting v from G destroys all s-t paths.

2. (25 points) Give an efficient algorithm that takes as input a directed graph $G = (V, E)$ and determines whether or not there is a vertex $s \in V$ from which all other vertices in the graph are reachable. Assume $|V| = n, |E| = m$.

3. (25 points) Give an efficient algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two vertices $s, t \in V$, and outputs the number of different paths from s to t in G. Assume $|V| = n, |E| = m$.

4. (35 points) You are given two bottles with capacities X and Y liters, respectively.

   Initially, there are x liters of water in the first bottle, and y liters of water in the second bottle, where $0 \le x \le X$ and $0 \le y \le Y$.

   At each step, you can perform one of the three operations below:

   - **FILLUP(i)**. Fill up bottle i with tap water.
   - **EMPTY(i)**. Pour all the water from bottle i to the drain.
   - **POUR(i, j)**. Pour the water from bottle i to bottle j. After this operation, either bottle i is empty or bottle j is full while there may be some water left in bottle i.

   Your goal is to end up with exactly A liters of water in one of the two bottles, for some $A \le \max\{X, Y\}$.

   Design an algorithm that, on input X, Y, x, y, A, achieves this goal by using the smallest number of operations, and returns that number. If it is not possible to achieve this goal, return "impossible" in your pseudo-code.

5. (35 points) In this problem, you will design a linear-time algorithm for `2SAT`.

   Consider a CNF formula ϕ with m clauses and n variables such that every clause consists of 2 literals. Given ϕ, construct a directed graph $G_\phi = (V, E)$ as follows:

   - G_ϕ has $2n$ nodes, one for each variable and its negation;
   - Note that a clause $(x \vee y)$ is equivalent to either of the implications $(\neg x \Rightarrow y)$ or $(\neg y \Rightarrow x)$. Introduce all these implications as edges of G_ϕ: for every clause $(x \vee y)$ add one directed edge from $\neg x$ to y and one directed edge from $\neg y$ to x.

   Show that ϕ is satisfiable if and only if there is no strongly connected component of G_ϕ that contains both a variable x and its negation $\neg x$. Then conclude that there is a linear-time algorithm for solving `2SAT`.

   Hint: First show that if G_ϕ has a strongly connected component containing both a variable x and its negation $\neg x$, then ϕ is not satisfiable. Then show the converse of this statement, (that is, if no strongly connected component of G_ϕ contains both a variable x and its negation $\neg x$, then ϕ is satisfiable) by exhibiting an appropriate satisfying truth assignment for ϕ.

RECOMMENDED EXERCISES (do NOT submit, they will not be graded)

1. Problem 22-2, parts a, b, c, d, e and f, in your textbook (p. 622).