CSOR W4231–Spring, 2023
Eleni Drinea

Homework 5 (140 points)

Out: Monday, April 10, 2023
Due: 11:59pm, Tuesday, April 25, 2023 — HARD DEADLINE

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, even if you don't give an entirely formal proof.
   - 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. 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 but you're encouraged to do so.)
   (c) Prove carefully equivalence of the original and the reduced instances.

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

4. 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.

5. **You should not use any external resources for this homework.** Failure to follow this instruction may have a negative impact on your performance in the second exam and interviews. For the same reason, **you should 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. I also encourage you to work on all the recommended exercises.

6. You should write up the solutions **entirely on your own**. Collaboration is limited to discussion of ideas only and 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. You should list your collaborators on your write-up.

Homework Problems

1. (40 points) A *flow network with demands* $G = (V, E, c, d)$ is a directed capacitated graph with potentially multiple sources and sinks, which may have incoming and outgoing edges respectively. In particular, each node $v \in V$ has an integer *demand* d_v; if $d_v > 0$, v is a *sink*, while if $d_v < 0$, it is a *source*. Let S be the set of source nodes and T the set of sink nodes.

   A *circulation with demands* is a function $f : E \to R^+$ that satisfies

   (a) *capacity constraints:* for each $e \in E$, $0 \leq f(e) \leq c_e$.

   (b) *demand constraints:* For each $v \in V$, $f^{\text{in}}(v) - f^{\text{out}}(v) = d_v$.

   We are now concerned with a decision problem rather than a maximization one: *is there* a circulation f with demands that meets both capacity and demand conditions?

   i. (10 points) Derive a necessary condition for a feasible circulation with demands to exist.

   ii. (30 points) Reduce the problem of finding a feasible circulation with demands to max ffow.

2. (45 points) In this problem, the input is a standard flow network $G = (V, E, s, t, c)$ with integer edge capacities.

   (a) (25 points) An edge in a flow network is called *critical* if decreasing the capacity of this edge results in a decrease in the value of the maximum flow.

   On input a flow network $G = (V, E, s, t, c)$, you ran Ford-Fulkerson and computed a maximum flow f.

   i. Give an efficient algorithm that finds a critical edge in G.

   ii. Give an efficient algorithm that finds every critical edge in G.

   (b) (20 points) Suppose we have already found the maximum s-t flow in G. However, we made a mistake in the capacity value of edge (u, v): we used c_{uv} but the capacity is only $c_{uv} - 1$. Moreover, the max flow f uses edge (u, v) at full capacity. Can you find a new optimal flow faster than by recomputing max flow in G?

3. (35 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.

   Here is a simple randomized algorithm that returns a truth assignment for the n variables.

```
for each variable do
    set its truth value to either 0 or 1 by flipping a coin
end for
```

   For $1 \leq j \leq m$, let k_j be the number of literals in the j-th clause has k_j literals You may assume that all variables in a clause are distinct.

   (a) (12 points) Give the expected number of clauses satisfied by the above algorithm in terms of m, n, k_j. Then lower bound this expectation in terms of m and n only.

(b) (23 points) Next, suppose we *de-randomize* the above algorithm as follows: instead of flipping a coin to assign a truth value to each variable, we assign to the variable the truth value that satisfies the most as-yet-unsatisfied clauses.

Give a lower bound for the number of clauses satisfied by this deterministic algorithm.

Hint: Let ℓ_i be the number of clauses containing at least one of the variables $\{x_1, \ldots, x_i\}$. Show that, for every i, after i variables have been assigned, the number of satisfied clauses is at least $\ell_i/2$.

4. (20 points) On the hardness of `MAX SAT`.

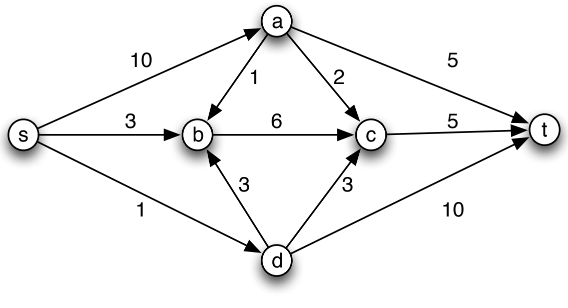
   (a) (5 points) Formulate the decision version of `MAX SAT` (see Problem 3 above).

   (b) (13 points) Prove that `MAX SAT(D)` is $\mathcal{NP}$-complete.

   (c) (2 points) Is the algorithm in Problem 1 an efficient algorithm? If your answer is yes, does this contradict what you proved in part b) of this problem? Explain your answer.

RECOMMENDED exercises: do NOT return, they will not be graded.)

1. Run the Ford-Fulkerson algorithm on the following network, with edge capacities as shown, to compute the max s-t flow. At every step, draw the residual graph and the augmenting paths. Report the maximum flow along with a minimum cut.



2. There are many variations on the maximum flow problem. For the following two natural generalizations, show how to solve the more general problem by **reducing** it to the original max-flow problem (thereby showing that these problems also admit efficient solutions).

   - There are multiple sources and multiple sinks, and we wish to maximize the flow between all sources and sinks.
   - Both the edges *and the vertices* (except for s and t) have capacities. The flow into and out of a vertex cannot exceed the capacity of the vertex.

3. 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.

4. *(Using reductions to prove $\mathcal{NP}$-completeness)*

   (a) 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. 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 the above maximization problem and show that it is $\mathcal{NP}$-complete. *Hint: reduction from* `Independent Set`.

   (b) We say that G is a *subgraph* of H if, by deleting certain vertices and edges of H we obtain a graph that is, up to renaming of the vertices, identical to G.

   The following problem has applications, e.g., in pattern discovery in databases and in analyzing the structure of social networks.

   `Subgraph Isomorphism:` Given two undirected graphs G and H, determine whether G is a subgraph of H and if so, return the corresponding mapping of vertices in G to vertices in H.

   Show that `Subgraph Isomorphism` is $\mathcal{NP}$-complete.

   (c) Similarly, consider the following problem.

   `Dense Subgraph:` Given a graph G and two integers a and b, find a set of a vertices of G such that there are at least b edges between them.

   Show that `Dense Subgraph` is $\mathcal{NP}$-complete.

5. 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 $V = \{1, \ldots, n\}$ 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 $j \in S$.