CSOR S4231 – Summer 2018

Homework 4

Out: Saturday, June 16, 2018

Due: 10pm, Tuesday, June 26, 2018

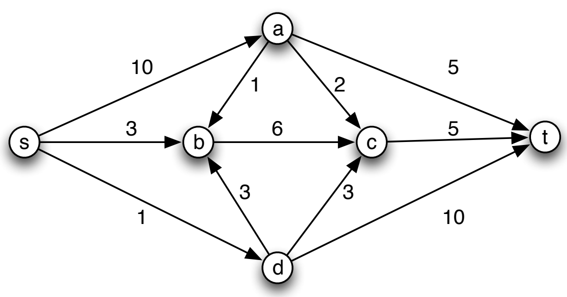
Please keep your answers clear and concise. For all algorithms **you** *suggest, you must prove correctness and give the best upper bound that you can for the running time. You should always describe your algorithm clearly in English* **and** *give pseudocode.*

You should work on all recommended exercises as they will help you prepare for the final exam.

You should write up the solutions entirely on your own. Collaboration with your classmates is limited to discussion of ideas only. You should list your collaborators on your write-up. If you do not type your solutions, make sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.

Notational convention: $|V| = n$, $|E| = m$.

1. (20 points) 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. (20 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 do not want to overload any single hospital; instead, you want every hospital to receive at most $\lceil n/k \rceil$ people.

   Give a polynomial-time algorithm that solves the above problem.

3. (30 points) In a *min-cost flow* problem, the input is a flow network with supplies as described in recommended exercise 2 where each edge $(i, j) \in E$ also has a cost a_{ij}. Thus this problem generalizes two important problems we discussed so far: max flow and shortest paths.

   Given a flow network with capacities, supplies and costs, the goal is to find a feasible flow $f : E \to R^+$ —that is, a flow satisfying edge capacity constraints and node supplies— that minimizes the total cost of the flow $\sum_{(i,j)\in E} a_{ij} f_{ij}$.

   (a) Formulate max flow as a min-cost flow problem.

   (b) Formulate the following problem as a min-cost flow problem.

   There are n people, n jobs and a set E of m ordered pairs of indices (i, j), indicating that person i qualifies for job j; associated with each pair (i, j) is a value v_{ij}, indicating how desirable job j is to person i. We want to find a subset A of the form $\{(1, j_1), (2, j_2), \ldots, (n, j_n)\}$ such that $j_1, j_2, \ldots, j_n$ are all distinct and the total value $\sum_{(i,j)\in A} v_{ij}$ is maximized.

4. (25 points) Consider an instance of the Satisfiability problem where all literals in every clause are unnegated variables. Such an instance is called a *monotone* instance. For example,

$$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_4)$$

   is a monotone instance. Monotone instances are trivially satisfiable: set *every* variable to 1 (true). However, the instance above would still be satisfied if, instead of setting $x_1 = x_2 = x_3 = x_4 = 1$, we only set $x_1 = x_2 = 1$.

   Show that the following decision problem is NP-complete: given a monotone instance of Satisfiability and a number k, is there a satisfying truth assignment for the instance in which at most k variables are true?

5. (25 points) Consider the following *restricted* version of `3SAT`: every literal appears at most *once*. Show that this problem can be solved in polynomial time.

   Hint: construct a bipartite graph.

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

1. 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.

2. A *flow network with supplies* is a directed capacitated graph with potentially multiple sources and sinks, which may have incoming and outgoing edges respectively. In particular, each node $i \in V$ has an integer *supply* s_i; if $s_i > 0$, i is a *source*, while if $s_i < 0$, it is a *sink*. Let S be the set of source nodes and T the set of sink nodes.

   A *circulation with supplies* 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) *supply constraints:* For each $i \in V$, $f^{\text{out}}(i) - f^{\text{in}}(i) = s_i$.

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

   i. Derive a necessary condition on the supplies s_i for a feasible circulation with supplies to exist.

   ii. Reduce the problem of finding a feasible circulation with supplies to Max Flow.

3. *(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.