Problem Set 5

Due: Friday, October 14, 2016.

Collaboration policy: collaboration is *strongly encouraged*. However, remember that

1. You must write up your own solutions, independently.
2. You must record the name of every collaborator.
3. You must actually participate in solving all the problems. This is difficult in very large groups, so you should keep your collaboration groups limited to 3 or 4 people in a given week.
4. **No bibles. This includes solutions posted to problems in previous years.**

Problem 1. The US Census Bureau produces a variety of tables from its Census data. Suppose that it wishes to produce a p by q table $D = \{d_{ij}\}$ of non-negative integers. Let r_i denote the sum of the matrix elements in the i^{th} row, and let c_j denote the sum of the elements in the j^{th} column. Assume that each sum r_i and c_j is strictly positive. The Census Bureau often wishes to disclose all the row and column sums along with some matrix elements (denoted by the set Y) and yet to suppress the remaining elements to ensure the confidentiality of privileged information. Unless it exercises care, by disclosing the elements in Y the Bureau might permit someone to deduce the exact value of one or more suppressed elements. It is possible to deduce the exact value of an element d_{ij} if only one value of d_{ij} is consistent with the row and column sums and the disclosed elements in Y. We say that any such suppressed element is *unprotected*. We wish to develop an efficient, flow-based algorithm for identifying all the unprotected elements of the matrix and their values.

(a) Given the table as described above, give an efficient, flow-based algorithm for deciding if there is *some* set of cell values that produces the disclosed values.

(b) Given a graph G with integer capacities and a max-flow f that places flow value f_e on edge e, give an efficient algorithm for determining whether there is a max-flow that places a *different* flow value on edge e.

(c) Combine the previous two parts to solve the stated problem.

NONCOLLABORATIVE Problem 2. Consider a unit-capacities network where each edge has cost 1. Does the shortest augmenting path algorithm (for max-flow, that just minimizes the number of edges on the path) or Dinic's normal blocking-flow algorithm compute a min-cost max-flow in the graph? Why or why not?

Problem 3. You are TAing a class with s students and r recitations. The students tell you which recitations they can fit into their schedules. However, each recitation has room for only k students. (Conversely, each student only has to attend one recitation.) Your goal is to assign the most students to recitations within these constraints.

(a) Suppose first that $k = 1$ (personal tutors!). How can you easily solve this problem in $O(m\sqrt{n})$ time?

(b) Now suppose k is arbitrary. How can you solve this problem with maximum flow?

(c) Argue that an $O(m\sqrt{n})$ time bound can be achieved for your network from part (b), as in unit networks.

Problem 4. You are given a graph G with integer capacities but arbitrary real costs, and a min-cost flow which is *not* integral. You will design a *strongly* polynomial algorithm to find an *integer* min-cost flow.

(a) Give an example of such a graph with some non-integral min-cost flow.

(b) Show that for any such flow, the residual graph must contain a cycle of edges of fractional residual capacity.

(c) What can you say about the cost of this cycle?

(d) Use the insight above to design a strongly polynomial algorithm to transform any non-integral flow into an integral one.

Problem 5. Every April, the MIT admissions office hosts several thousand "prefrosh" admitted students. They need to assign each prefrosh to an MIT-student host. Students and prefrosh fill out a multiple-choice "profile" that the admissions office uses to automatically compute a "suitability" function $f(x, y)$. Sometimes $f(x, y) = -\infty$ due to allergies or other issues. We were asked to find an optimal (maximum total suitability) assignment of prefrosh to student hosts, subject to the following constraints:

- No prefrosh can be assigned to a student if their suitability is $-\infty$.
- Each student can be assigned to at most one prefrosh, and each prefrosh must be assigned to exactly one student.
- Students occupy suites of possibly multiple students (each student in exactly one), and each suite s has room for only some maximum number m_s of prefrosh.
- Suites are spread among floors of dormitories, and fire code limits the total number of prefrosh on floor g to some number M_g. Assume that each suite is on only one floor.
- Similarly, there is a total fire-code limit $\mathcal{M}_d$ on the number of prefrosh who should be assigned to each dormitory d.

(a) Assuming there is a feasible solution, show how the problem can be solved by an application of min-cost flow.

(b) If there is no feasible solution, it may be necessary to break the limits on suites (but the fire-code is inflexible). Give an efficient algorithm that finds the solution that minimizes the total “overage” (sum of amounts by which individual suite limits m_s are broken) and, among such solutions, maximizes the total suitability. Your algorithm should detect if there is no such solution.

Problem 6. How long did you spend on this problem set? Please answer this question using the Google form that is located on the course website. This problem is mandatory, and thus counts towards your final grade. It is due by the Monday 2:30pm after the pset due date.

Optional problems on Spectral Graph Theory

The following problems are about the material covered in the optional lecture on Monday 10/3. Recall there is another upcoming optional lecture on October 12 that will continue this one.

For all the following problems, let $G(V, E)$ a simple, unweighted and undirected graph, with $|V| = n$, and Laplacian matrix L.

OPTIONAL Problem 7. Let G be a graph with Laplacian eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. In class we proved that G is connected iff $\lambda_2 > 0$. Prove the following generalization: The number of connected components in G is equal to the multiplicity of 0 as a Laplacian eigenvalue (i.e. the number of λ_i's that equal 0).

OPTIONAL Problem 8. Suppose G is d-regular, meaning all vertices have the same degree d. In class we showed that if G is bipartite, then $\lambda_n = 2d$.

(a) Prove the converse: If G is connected and $\lambda_n = 2d$, then G is bipartite.

(b) Does this statement hold if G is not connected?