CS 38 Introduction to Algorithms **Spring 2014**

Problem Set 4

Out: May 6 *Due:* **May 13**

Reminder: you are encouraged to work in groups of two or three; however you must turn in your own write-up and note with whom you worked. You may consult the course notes and the optional text (CLRS). The full honor code guidelines can be found in the course syllabus.

Please attempt all problems. **To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.**

1. Consider the problem of correcting typing errors that omit spaces; for example we would like to correct "meetatone" to "meet at one" and not "me eta tone" or many others decompositions that are less plausible. Assume that we have access to a quality function q that returns a numerical score for each string. The quality of a decomposition of the string $x = x_1 x_2 \cdots x_n$ into $x = y_1 y_2 \cdots y_k$, where each y_i is a string itself, is given by $\sum_{i=1}^{k} q(y_i)$.

   Devise a dynamic programming algorithm that computes the highest quality decomposition of x. Your algorithm should run in time $O(n^3)$, counting evaluations of q as unit cost operations.

2. Given a directed graph G with positive edge weights, and two vertices s and t, we would like to compute a measure of the "robustness" of the shortest path from s to t, by computing the *number* of these shortest paths.

   Devise a dynamic programming algorithm to compute the number of shortest s-t paths in G, that runs in time $O(nm)$. You may count arithmetic operations as unit cost operations.

3. Let E be a finite set and let $\mathcal{I}_1$ and $\mathcal{I}_2$ be matroids over E. The problem of finding a maximum *cardinality* independent set in $\mathcal{I}_1 \cap \mathcal{I}_2$ can be solved in polynomial time, with polynomially many checks to determine if a specified subset is an indepedent set in one of the two matroids.

   Formulate the bipartite maximum matching problem as the problem of finding a maximum cardinality independent set in the intersection of two matroids.

4. In this problem you will prove two facts about the structure of perfect matchings in bipartite graphs.

   (a) A k-regular graph is one in which the degree of every vertex is k. Prove that every k-regular bipartite graph has a perfect matching by formulating the matching problem as a flow problem and studying the maximum flow.

   (b) A more general fact is known as Hall's Theorem. Let $G = (U \cup V, E)$ be a undirected bipartite graph, and for a subset $S \subseteq U$, let $N(S)$ denote the neighbors of the vertices in subset S.

Prove that there exists a perfect matching in G if and only if we have $|N(S)| \geq |S|$ for all subsets $S \subseteq U$. Hint: for one direction, formulate the matching problem as a flow problem, and use the min-cut to identify a subset S for which $|N(S)| < |S|$.

5. Extensions to max-flow and an application. Given a directed graph $G = (V, E)$, non-negative integer capacities $c(e)$ for each edge $e \in E$, and integer *demands* $d(v)$ for each vertex, a *circulation* is a function f satisfying

- $0 \leq f(e) \leq c(e)$ for all $e \in E$, and
- for all $v \in V$,
$$\sum_{e \text{ entering } v} f(e) - \sum_{e \text{ exiting } v} f(e) = d(v).$$

(a) Show how to find a circulation, or determine that one does not exist, by reducing to max-flow. Your algorithm should run in time comparable to the algorithms for max-flow on graphs with size parameters $n = |V|$ and $m = |E|$.

(b) Now suppose that in addition to a capacity $c(e)$, each edge e has a *lower bound* $\ell(e)$ and we are seeking a circulation f that satisfies $f(e) \geq \ell(e)$ for all $e \in E$. Show how to find a such a circulation, or determine that one does not exist, by reducing to max-flow. Your algorithm should run in time comparable to the algorithms for max-flow on graphs with size parameters $n = |V|$ and $m = |E|$.

(c) Survey design. Suppose we have n_1 consumers and n_2 products, and we are given a bipartite graph $G = (C \cup P, E)$ with an edge (c, p) to indicate that consumer c owns product p. We want to perform a consumer satisfaction survey in which consumer c is asked between q_c and q'_c questions and product p is surveyed between s_p and s'_p times. We can only ask consumer c about product p if they own it.

Given G, values q_c, q'_c for each $c \in C$, and s_p, s'_p for each $p \in P$, design an algorithm to determine whether the specified survey parameters are feasible, by formulating this as the problem of finding a circulation with lower bounds.