CMPSCI 311: Introduction to Algorithms	**Fall 2016**

Homework 5

Released 11/4/2016 — Due 8:00pm 11/11/2016 in Moodle

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Instructions. You may work in groups, but you must individually write your solutions yourself. List your collaborators on your submission.

If you are asked to design an algorithm as part of a homework problem, please provide: (a) the pseudocode for the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm, and (e) justification for your running time analysis.

Submission instructions. This assignment is due by 8:00pm on 11/11/2016 in Moodle. Please submit a pdf file. You may submit a scanned handwritten document, but a typed submission is preferred.

1. **Sequence alignment (20 points).** This problem considers the sequence alignment problem with a twist. As before, we are given two strings x, y where $\texttt{length}(x) = m$ and $\texttt{length}(y) = n$, and a cost matrix C which prescribes a non-negative cost for matching characters. Instead of a gap penalty δ, we are given a gap penalty *function* $f : \mathbb{N} \to \mathbb{R}$ that gives cost $f(i)$ when i consecutive characters are x are matched with gaps or when i consecutive characters of y are matched with gaps.

   This is a general version of our original sequence alignment problems which has $f(i) = i \times \delta$ for all i.

   For this problem, assume that $\frac{f(1)}{1} \geq \frac{f(2)}{2} \geq \ldots \frac{f(n)}{n} \ldots$ so that the sequence $\frac{f(i)}{i}$ is non-increasing. This means that the marginal cost of a gap is *less*, if it is in a long segment of gaps.

   Example. Consider the alignment

   ```
   a  b  c  d  -  e  -  -
   |  |  |  |  |  |  |  |
   -  b  f  -  h  -  i  j
   ```

   This alignment has cost $C(b, b) + C(c, f) + f(1) + f(2) + f(3)$, since there are gap sequences of length one ("a"), two ("de"), and three ("hij").

   Design an efficient algorithm for finding the alignment of minimal cost under this new criteria.

2. **Disaster Management, K&T Ch.7 Ex.9 (15 points).** A natural disaster has hit the Pioneer Valley! The paramedics have n injured people that they need to get to k hospitals. Each person needs to get to a hospital within a half-hour driving time from their current location, so each person i has a set of hospitals they can go to $S_i \subset \{1, \ldots, k\}$. At the same time, the paramedics do not want to overload any hospital, so they are not allowed to send more than $\lceil n/k \rceil$ patients to any single hospital.

   Design an algorithm that the paramedics can use to decide whether it is possible to get injured people to the hospital, so that both the locality constraint and the overloading constraint are satisfied.

   Your algorithm need not output a assignment of people to hospitals, just whether one exists or not.

3. **Network Flows, K&T Ch.7 Ex.12 (15 points).** You are given a flow network with unit-capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter k. The goal is to delete k edges so as to reduce the maximum $s - t$ flow in G by as much as possible. In other words, you should find a set of edges $F \subset E$ so that $|F| = k$ and the maximum $s - t$ flow in $G = (V, E \setminus F)$ is as small as possible subject to this. Give a polynomial-time algorithm to solve this problem.

4. **(0 points).** How long did it take you to complete this assignment?