Graph Algorithms (part 2 of CSC 282),

http://www.cs.rochester.edu/~stefanko/Teaching/17CS282

IMPORTANT: **Write each problem on a separate sheet of paper. Write your name on each sheet.** (Why? To speed up the grading we will use parallelization—each problem will be graded by a different TA. At the beginning of the class (on the date when the homework is due) there will be a separate pile for each problem.)

Homework 2 due: Oct. 5 (Thursday); collected before class in Wegmans 1400.

Problem sessions before Homework 2 is due:

Oct. 2 (Monday) 6:15pm - 7:15pm in Hylan 202
Oct. 3 (Tuesday) 6:15pm - 7:15pm in Goergen 109
Oct. 4 (Wednesday) 6:15pm - 7:15pm in Gavett 310
Oct. 4 (Wednesday) 7:40pm - 8:40pm in Hutchinson 473

1 Homework - solve and turn in

2.1 (**due Thursday, Oct. 5**) Let $G = (V, E)$ be a directed graph (digraph) given in the following adjacency-list representation: for each vertex $v \in V$ we have a linked list of out-neighbors of v. Assume that $V = \{1, \ldots, n\}$ and let $m := |E|$.

1. Write pseudocode for a procedure which outputs an adjacency-list representation of the reverse digraph (i. e., G with each edge reversed). The procedure should run in time $O(m + n)$.
2. Write pseudocode for a procedure which outputs the adjacency-list representation of G in which the out-neighbors of each vertex are listed in the increasing order. The procedure should run in time $O(m + n)$.
3. Write pseudocode for a procedure which checks if G is undirected (i. e., the reverse of every $e \in E$ is also in E). The procedure should run in time $O(m + n)$.

2.2 (**due Thursday, Oct. 5**)

1. We are given a DAG (directed acyclic graph) $G = (V, E)$ and two vertices $u, v \in V$. Assume that $V = \{1, \ldots, n\}$ and let $m := |E|$. We want to figure out the length of the longest u-v-path in G (the length of a path $v_1, \ldots, v_k$ is $k - 1$). Give an $O(m + n)$ algorithm for the problem. Write pseudocode for the algorithm.
2. We are given a DAG $G = (V, E)$, two vertices $u, v \in V$ and a subset $S \subseteq V$. We want to figure out whether there exists a u-v-path in G that contains a vertex in S (that is, does there exist k and a path $v_1, \ldots, v_k$ such that $v_1 = u$, $v_k = v$ and $\{v_1, \ldots, v_k\} \cap S \neq \emptyset$). Give an $O(m+n)$ algorithm for the problem. Write pseudocode for the algorithm.
3. We are given a directed graph $G = (V, E)$, two vertices $u, v \in V$ and a positive integer ℓ. We want to figure out whether there exists a u-v-walk of length **at least** ℓ (remember, in a walk we allow repeated vertices and edges; the length of a walk $v_1, \ldots, v_k$ is $k - 1$). Give an $O(m + n)$ algorithm for the problem. Clearly describe your algorithm (pseudocode is not required). You can use any algorithm covered in class/textbook (you don't need to write pseudocode for the algorithm used).

2.3 (**due Thursday, Oct. 5**) We have a road network consisting of one-way streets. There are a few proposals for a new one-way road to be built. You want to vote for the proposal that would result in the shortest distance between your home and your office. How can you, quickly, figure out the proposal to vote for? We now formalize the problem.

Let $G = (V, E, w)$ be a directed graph with positive edge weights; let $n := |V|$, $m := |E|$. Let $s, t \in V$ be two distinct vertices (s is your home, t is your office). There are k proposals $(a_1, b_1, c_1), \ldots, (a_k, b_k, c_k)$, where $a_i, b_i \in V$ and $c_i \in \mathbb{R}_+$ ((a_i, b_i, c_i) proposes connecting location a_i to location b_i with a road of length c_i).

Let G_i be the graph obtained from G by adding edge (a_i, b_i) with weight c_i (note that G_i has $m + 1$ edges (for each $i \in \{1, \ldots, k\}$)). We want to find which of $G_1, \ldots, G_k$ has the shortest distance from s to t. Describe an $O(m + n \log n + k)$ algorithm for the problem (no pseudocode is needed, just a clear description of the algorithm; you can use any algorithm that was mentioned in the class or in the book).

2 Bonus Homework - solve and turn in

2.4 (**due Thursday, Oct. 5**) Consider the same problem as 2.3 except now ℓ out of the k proposals get built. How can you, quickly, figure out the proposals to vote for?

Let $G = (V, E, w)$ be a directed graph with positive edge weights; let $n := |V|$, $m := |E|$. Let $s, t \in V$ be two distinct vertices (s is your home, t is your office). There are k proposals $(a_1, b_1, c_1), \ldots, (a_k, b_k, c_k)$, where $a_i, b_i \in V$ and $c_i \in \mathbb{R}_+$ ((a_i, b_i, c_i) proposes connecting location a_i to location b_i with a road of length c_i). Let $\ell \in \{1, \ldots, k\}$. We want to find $S \subseteq \{1, \ldots, k\}, |S| = \ell$ that minimizes the distance from s to t in the graph G with edges $(a_i, b_i, c_i), i \in S$ added in.

Describe a fast algorithm for the problem (no pseudocode is needed, just a clear description of the algorithm; you can use any algorithm that was mentioned in the class or in the book). Determine the asymptotic running time of your algorithm.

3 Additional problems

Try to solve the following problems from Cormen-Leiserson-Rivest-Stein. We will go over the ones that you choose in the problem sessions.

- 22.1-1, 22.1-6, 22.1-7, 22.2-6, 22.2-8, 22.3-9, 22.3-11, 22.5-1, 22.5-3, 22.5-4, 22.5-7,
- 24.1-5, 24.1-6, 24.3-7, 24.3-8, 24.3-9, 24.4-5,
- 25.2-6, 25.2-8,
- 26.2-10, 26.2-11.

Try to solve the following problems from Dasgupta-Papadimitriou-Vazirani. We will go over the ones that you choose in the problem sessions.

- 3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8, 3.9, 3.11, 3.12, 3.13, 3.15, 3.18, 3.19, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27,
- 4.1, 4.2, 4.3, 4.4, 4.5, 4.7, 4.8, 4.10, 4.11, 4.13, 4.15, 4.17, 4.18, 4.19, 4.20, 4.21,
- 5.1, 5.2, 5.3, 5.7.

Try to solve the following problems from `http://jeffe.cs.illinois.edu/teaching/algorithms/all-graphs.pdf` We will go over the ones that you choose in the problem sessions.

- Section 18: problems 9, 10, 11, 12,
- Section 19: problems 1, 5, 6, 7, 9,
- Section 20: problems 1, 2, 3, 4, 5, 6, 7, 8,
- Section 21: problems 4, 5, 6, 7, 9, 11, 12,
- Section 2: problems 3, 4, 5, 6.