CS420/520: Graph Theory with Applications to CS, Winter 2016

Homework 1

Due: Tue, Jan/19/16

Homework Policy: Each student should submit his/her own set of solutions, independently. You are allowed to discuss the homework with other students, however, you need to indicate their names in your submission.

Readings:

(A) Jeff lecture notes on basic graph algorithms: http://jeffe.cs.illinois.edu/teaching/algorithms/notes/18-graphs.pdf.

(B) Jeff lecture notes on graph search: http://jeffe.cs.illinois.edu/teaching/algorithms/notes/19-dfs.pdf.

(C) Jeff lecture notes on single source shortest paths: http://jeffe.cs.illinois.edu/teaching/algorithms/notes/21-sssp.pdf.

Problems for practice.

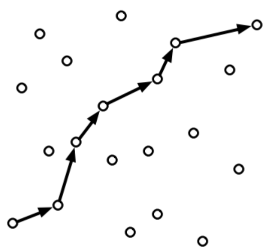
- Problems (1), (3), (7), (12) from (A).
- Problems (7), (8) from (B).
- Problem (8) from (C).

Problem 1. A graph $G = (V, E)$ is bipartite if the vertices V can be partitioned into two subsets L and R, such that every edge has one vertex in L and the other in R.

(a) Prove that every tree is a bipartite graph.

(b) Describe and analyze an efficient algorithm that determines whether a given undirected graph is bipartite.

Problem 2. A polygonal path is a sequence of line segments joined end-to-end; the endpoints of these line segments are called the vertices of the path. The length of a polygonal path is the sum of the lengths of its segments. A polygonal path with vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ is monotonically increasing if $x_i < x_{i+1}$ and $y_i < y_{i+1}$ for every index i (informally, each vertex of the path is above and to the right of its predecessor). Here is an example of a monotone path.



Suppose you are given a set S of n points in the plane, represented as two arrays $X[1 \ldots n]$ and $Y[1 \ldots n]$. Describe and analyze an algorithm to compute the length of the maximum-length monotonically increasing path with vertices in S. The length of a path is the number its vertices.

Problem 3. Prove that the following invariant holds for every integer i and every vertex v: After i phases of Shimbel's algorithm (in either formulation), $dist(v)$ is at most the length of the shortest path from s to v consisting of at most i edges.