Problem Set 5
CSE 373 Fall 2015
Due 12/7 2015

Notes on Grading

• You can write “I don’t know” for any question and receive 25% credit. You can take this option for any numbered problem, but not for part of a problem. For example, you can answer 3.1 and write “I don’t know” for 3.2, but you can’t write part of the solution for 3.2 and then write “I don’t know” for the rest.

• You get a 10% bonus for typing your homework. You are encouraged to use \LaTeX. You must type your entire homework to receive the bonus. The 10% bonus does not apply to problems answered with “I don’t know.”

1 DFS

The edges of an undirected graph with vertexes $v_1$ to $v_7$ are represented by the following the adjacency matrix:

$$
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

(The adjacency matrix $A$ of a graph has $A_{ij} = 1$ iff there is an edge from vertex $v_i$ to vertex $v_j$)

(5 points) Starting with $v_1$, run DFS and draw the resulting DFS tree. (The DFS tree is the set of edges that DFS follows to vertices that it hasn’t already visited.)
2 Dijkstra’s Algorithm

(5 points) Run Dijkstra’s algorithm on the weighted graph below, using vertex A as the source. Write the vertices in the order which they are marked.

![Graph Diagram]

3 Connected Components

Suppose you are also given a look-up table T where $T[u]$ for $u \in V$ is a list of guests that $u$ knows. If $u$ knows $v$, then $v$ knows $u$.

You are required to arrange the seating such that any guest at a table knows every other guest sitting at the same table either directly or through some other guests sitting at the same table. For example, if $x$ knows $y$, and $y$ knows $z$, then $x$, $y$, $z$ can sit at the same table.

(10 points) Describe an efficient algorithm that, given $V$ and $T$, returns the minimum number of tables needed to achieve this requirement. Analyze the running time of your algorithm.

4 Max-flow min-cut

Figure 1 shows a flow network with the capacity on each edge.

- (5 points) Find a maximum $s$-$t$ flow (by drawing the flow on the network and indicate the value of the flow).
- (5 points) Find a minimum $s$-$t$ cut (by specifying the vertices to sets of the cut).
- (5 points) Draw the final residual network and the edge capacities. In the residual network, mark the vertices reachable from $s$ and the vertices from which $t$ is reachable.
• (5 points) An edge is called **upward critical** if increasing the capacity of an edge increases the maximum flow. Find an upward critical edge of Network I if one exists.

• (5 points) An edge is called **downward critical** if decreasing the capacity of the edge decreases the maximum flow. Find an downward critical edge of Network I if one exists.

• (10 points) Suppose we have a max flow $f$ of a given network $N = \{G = \{V, E\}, c, s, t\}$. At a time the capacity of one edge $(u, v) \in E$ is increased by 1 to form a new network $N'$. Can you design an efficient ($O(|V| + |E|)$-time) algorithm to update the max flow $f'$ of $N'$?