CSE 3101Z Design and Analysis of Algorithms
Professor: James Elder
Winter 2009
Assignment 3
Due 11:59pm Monday May 4

First Person:                                           Second Person:
Family Name:                                           Family Name:
Given Name:                                            Given Name:
Student #:                                             Student #:
Email:                                                 Email:

Guidelines:

• You may complete this assignment alone or in groups of two. Do not get solutions from other pairs. Though you are to teach & learn from your partner, you are responsible to do and learn the work yourself. Write it up together.

• Please make your answers clear and succinct.

• Relevant Readings:
  – CLRS Ch. 22-26, B
  – Edmonds Ch. 14-15

• This page should be the cover of your assignment.

• New: Answer Questions 1 and 2 directly on the assignment sheets and hand in these two sheets as the 2nd and 3rd pages of your assignment.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (10 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 (10 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 (15 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (20 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 (25 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 (20 marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (100 marks)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (10 marks) **Breadth- and Depth-First Search**

Trace BFS and DFS on the following two graphs. Hand in this page as Page 2 of your assignment. For each do the following:

(a) (2 marks) Start at node $s$ and when there is a choice follow edges from left to right. **Number** the nodes 1, 2, 3, ... in the order that they are found, starting with Node $s = 1$.

(b) (2 marks) **Darken** the edges of the Tree specified by the predecessor array $\pi$.

(c) (1 mark) What is the data structure used by BFS to store nodes that are found but not yet handled?

(d) (2 marks) Circle the nodes that are in this data structure when Node 8 is first found.

(e) (1 mark) What is the data structure used by DFS to store nodes that are found but not yet handled?

(f) (2 marks) Circle the nodes that are in this data structure when Node 8 is first found.
2. (10 marks) **Dijkstra’s Algorithm.** Hand in this page as Page 3 of your assignment.

(a) (5 marks) Consider running Dijkstra’s algorithm on the following graph from source node \( a \). Suppose we are at the stage of the computation when only nodes \( a \), \( b \) and \( d \) have been handled (in other words, we are just about to dequeue another node). Please indicate on the graph the current value of the distance estimate \( d \) for each node in the graph.

(b) (5 marks) Now consider running the next iteration of Dijkstra’s algorithm on this graph. On the graph below, show the new values of \( d \) for each node and any *changed* values of the predecessor array \( \pi \).
3. (15 marks) **Network flow**

Consider the flow network shown below, with source $s$ and terminal (or sink) $t$. The labels on the edges are of the form $f/c$ where $f$ is the flow and $c$ is the capacity of the edge. This flow is feasible.

Consider one iteration of the network flow algorithm (i.e., the Ford-Fulkerson algorithm) on this flow network.

(a) (5 marks) Show the residual network for this flow. Identify an augmenting path on this network.

(b) (3 marks) Draw the new flow network obtained by adding the flow along this path.

(c) (2 marks) What is the value of this flow?

(d) (5 marks) Prove that this flow is a maximum flow.

4. (20 marks) **Fixed edge on a shortest path** Given a directed, weighted graph $G$ with non-negative weights (weights can be zero), vertices $s, t$ and an edge $(u, v)$ in $G$

(a) (10 marks) Describe in concise English how to determine whether $(u, v)$ occurs on every path of minimal cost from $s$ to $t$.

(b) (10 marks) Describe in concise English how to determine whether $(u, v)$ occurs on some path of minimal cost from $s$ to $t$.

5. (25 marks) **Greenhouses and plants** A botanical garden is planning to build $n$ greenhouses, each of which is intended to represent a different climate zone. They have a list of $m$ kinds of plants that they would like to grow there. Not every kind of plant can grow in every greenhouse, though usually one kind can grow in more than one climate; so associated with every greenhouse $i \in [1 \ldots n]$ is a list $C_i$ of plant varieties it can grow. In addition to that, every greenhouse can host no more than 20 varieties of plants.

(a) (15 marks) Given the $n, m$ and $C_i$ for $1 \leq i \leq n$, describe in concise English an algorithm to determine the maximal number of different varieties of plants that can be grown in these greenhouses (total over all greenhouses).
(b) (10 marks) How can you produce a list of chosen varieties for each greenhouse? Is this list uniquely determined? Justify your answer with an example or a proof.

6. (20 marks) **Expedition** A group of people is planning an expedition. For safety, they decide to split into two groups; it does not matter how many people are in each group. Not all of them are good friends, and one reason for splitting is to separate all pairs that are not friendly to each other. Your goal is to design an algorithm that, given the map of friendship relations between people in the group, would find a partition of people into two groups of friends, or say that it is impossible.

Assume that the input is an $n \times n$ matrix $M$, where $n$ is the number of people in the group and $M(i,j) = 1$ if $i$ and $j$ are friends, and 0 otherwise. Also, if $i$ is friendly to $j$, then $j$ is friendly to $i$.

Your output should be the sequence of numbers corresponding to people assigned to the first group.

(a) (10 marks) Describe in concise English an algorithm that solves this problem.

(b) (5 marks) Provide a brief argument for why the algorithm finds a correct solution.

(c) (5 marks) Provide tight bounds on best- and worst-case running times.