Reminders

Remember:

- Write your NetID (your email ID) on the first page of your homework.
- Staple your pages - don’t just fold them together.
- Notice the due date - your homework is due at 2pm (when class begins), not later, after class has begun.

If you don’t follow these rules, expect to lose points!

1 Dijkstra’s Algorithm

Look at the graph later in this document. Using it, calculate the shortest path from node A to all of the other nodes. Show your work by filling out a table, showing each step of the algorithm. On each row, show the current best estimates to each node.

For nodes that you have not yet found any path, leave the space blank.

If you have found the final answer for a node, circle the answer - and then you don’t need to write it anymore.

I’ve given you the first 3 rows as an example:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td></td>
<td></td>
<td>1</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(best)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>11</td>
<td>1</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(best)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Why isn’t there a column J in the table? Because, when I randomly generated the graph, J never showed up. But I didn’t notice that until after I’d posted the graph.

After you have finished this table, run Dijkstra’s algorithm twice more: starting at the nodes E, and then C. But for these ones, you don’t have to draw out the entire table; instead, just give the last answer for each node.
Prim’s Algorithm

Using the same graph, run Prim’s Algorithm, starting at F. Use a table like you did for Dijkstra’s Algorithm: each row of the table represents one moment in time, and each entry in that row represents the best edge that you’ve found to reach various nodes.

As before, I’ll give you the first 3 steps.

After you have finished the algorithm, draw the spanning tree that you have chosen. (I’ll post the .dot file for the graph online, along with this homework.)

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>K</th>
<th>In the tree:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6</td>
<td>10</td>
<td></td>
<td>10</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>6</td>
<td>10</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F,A</td>
</tr>
</tbody>
</table>

**FIX:** I corrected the 3rd line on the table above.
3 Radix Sort

Remember Radix Sort (slide deck 7)?

In Radix Sort, we iteratively run Bucket Sort, using a different part of the key in each pass. In the first pass, we sort by the least important part of the key; in the last pass, we sort by the most important part.

3.1 Doing it by Hand

Use Radix Sort to sort the following four-letter words. Show the contents of the array after each iteration of Bucket Sort.

Make sure to check your work - if you sort it properly, then the last version of the array will be completely sorted!

alms
best
afar
burp
ball
anon
bags
cool
bell
coal
curl
adds
barn
bolt
case
bark