Goals

• Analyze graph search algorithms
CS200 - Problem Set 9
Due: Monday, April 30 to Canvas

1. Graph Search. In this problem we will consider the graph described by the following adjacency list:

<table>
<thead>
<tr>
<th>vertex</th>
<th>adjacency list</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>u, y</td>
</tr>
<tr>
<td>u</td>
<td>v, z, s</td>
</tr>
<tr>
<td>y</td>
<td>s, z</td>
</tr>
<tr>
<td>v</td>
<td>w, a, u</td>
</tr>
<tr>
<td>z</td>
<td>w, u, y</td>
</tr>
<tr>
<td>a</td>
<td>v, t</td>
</tr>
<tr>
<td>w</td>
<td>v, t, z</td>
</tr>
<tr>
<td>t</td>
<td>a, w</td>
</tr>
</tbody>
</table>

We will also use the algorithm DepthFirstSearch, which is a version of the Graph Search algorithm we saw in class. Its pseudocode is:

Algorithm 1: DepthFirstSearch(A, X, s, f)
Input: Adjacency list A for a graph G = (V, E), an array X of length |V| such that X[v] = 1 if v has been explored and 0 otherwise, a starting vertex s, a goal vertex f.
Output: String "f found!" or "f not found" depending on whether f can be found from s.

1. if s == f then
   2. Return "f found!";
1. else
   4. X[s] = 1;
   5. d = A[s].length;
   6. for k = 1 to d do
      7. if X[A[s, k]] == 0 then
         8. DepthFirstSearch(A, X, A[s, k], f);
     end
   end
   10. end
11. end
12. Return "f not found";

(a) [6 points] Please draw the graph the adjacency list corresponds to.

(b) [6 points] Suppose you start at s, and want to find t. In what order are vertices explored if you use DepthFirstSearch(A, X, s, t)? (Where X is initially set to all zeros)

(c) [6 points] Suppose you start at s, and want to find y. In what order are vertices explored if you use DepthFirstSearch(A, X, s, y)? (Where X is initially set to all zeros).

Provide the sequence of visited vertices if we use Breadth First Search on this graph, starting at s, and use the order given by the adjacency lists.
1. Graph Search. In this problem we will consider the graph described by the following adjacency list:

<table>
<thead>
<tr>
<th>vertex</th>
<th>adjacency list</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>u, y</td>
</tr>
<tr>
<td>u</td>
<td>v, z, s</td>
</tr>
<tr>
<td>y</td>
<td>s, z</td>
</tr>
<tr>
<td>v</td>
<td>w, a, u</td>
</tr>
<tr>
<td>z</td>
<td>w, u, y</td>
</tr>
<tr>
<td>a</td>
<td>v, t</td>
</tr>
<tr>
<td>w</td>
<td>v, t, z</td>
</tr>
<tr>
<td>t</td>
<td>a, w</td>
</tr>
</tbody>
</table>

We will also use the algorithm DepthFirstSearch, which is a version of the Graph Search algorithm we saw in class. Its pseudocode is:

```plaintext
Algorithm 1: DepthFirstSearch(A, X, s, f)
Input: Adjacency list A for a graph G = (V,E), an array X of length |V| such that X[v] = 1 if v has been explored and 0 otherwise, a starting vertex s, a goal vertex f
Output: String "f found!" or "f not found" depending on whether f can be found from s.
1) if s == f then
2) Return "f found!";
3) else X[s] = 1;
4) d = A[s].length;
5) for k = 1 to d do
6) if X[A[s,k]] == 0 then
7) DepthFirstSearch(A, X, A[s,k], f);
8) end
9) end
10) end
11) end
12) Return "f not found"
```

(a) [6 points] Please draw the graph the adjacency list corresponds to.

(b) [6 points] Suppose you start at s, and want to find t. In what order are vertices explored if you use DepthFirstSearch(A, X, s, t)? (Where X is initially set to all zeros)

(c) [6 points] Suppose you start at s, and want to find y. In what order are vertices explored if you use DepthFirstSearch(A, X, s, y)? (Where X is initially set to all zeros)?

Provide the sequence of visited vertices if we use Breadth First Search on this graph, starting at s, and use the order given by the adjacency lists.

s, u, y, v, z, w, a, t