

CS350 Homework 3

Russell Miller

February 27, 2011

22.1-1

The out-degree of a vertex is just its adjacency list, so in $\Theta(V)$ time you can get the out-degree of every vertex. Finding the in-degree for every vertex would require looking for that vertex in every vertex's adjacency list. So the entire adjacency list must be visited, which would be $\Theta(V * E)$, where V is the total number of vertices, and E is the total number of edges.

22.1-3

Adjacency List

```
[(1, [2,3]),  
 (2, [4,5]),  
 (3, [6,7])]
```

Adjacency Matrix

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	1	1	0	0
3	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

22.1-3

TRANSPOSE(ajd_list A)

```
1 new adj_list B  
2 for each v in A  
3   for each (v,x) in v  
4     B.add(x,v)
```

TRANSPOSE(adj_matrix A)

```
1 new ajd_matrix B[A.width][A.height]  
2 for i in A.width  
3   for j in A.height  
4     B[j][i] = A[i][j]
```

The running time of the list version of the transpose is $O(|V|^2)$, where V is the number of vertices. In the best case, with a completely disconnected graph, it would only take $|V|$.

The running time of the matrix version is $\Theta(|V|^2)$, because it copies the entire matrix which is width and height V .

22.1-6

Finding a universal sink has 3 parts: identifying the correct vertex, verifying that its in-degree is $|V| - 1$, and verifying that its out-degree is 0.

In order to find a vertex with zero out-degree, the worst-case would be $|V|^2$, so immediately we know this is not where to start. We also know that determining a vertex's in-degree is $|V|^2$.

So as a shortcut, we'll start from an arbitrary row in the matrix, and when a 1 is found in column j , move to row i , where $i = j$. Then find the first 1, repeating this process until a row i is found with all zeros. This could potentially be the universal sink. The last step is to verify that in column j , where $j = i$, there are all 1s.

While jumping from row to row, there were a few traversals within each row, for less than V rows. There was also the traversal of the sink row, which is length V . And lastly, there was the traversal of the V -length column. This is $3V$, which is $O(V)$.

This is clever, but it's not actually complete. This is because there are cases when this would be much worse than $O(V)$. Here is the matrix to show a counterexample. Imagine a graph where all vertices are connected to each other, AND the sink, and the algorithm traverses every edge before reaching the sink. It would take $O(V^2)$.

	1	2	3	4	5	6	7
1	0	1	0	0	0	0	1
2	0	0	1	0	0	0	1
3	0	0	0	1	0	0	1
4	0	0	0	0	1	0	1
5	0	0	0	0	0	1	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0