Brandon Toner

September 16, 2021

CSCI 405: Algorithm Analysis II

Homework 3: Elementary Graph Algorithms

1. (a) AllIn(A)

```
Data: A : adjacency matrix
Result: vertex index, or -1 if no result
$h = 0$;
while $h < |v|$ do
    $found = true$;
    for $k : h$ to $|V|$ do
        if $(k \neq h)$ AND $(A[k, h] \neq 0$ OR $A[h, k] \neq 1)$ then
            if $k > h$ then $h = k$ ;
            else $h = h + 1$;
            found = false;
            break;
        end
    end
    if $found$ then return $h$;
end
return $-1$
```

   (b) When a vertex satisfies the conditions, there is an edge from every other vertex to it, and there are no edges leading away from it. If there are two such vertices, then they would each have an edge leading from the other to them, which would mean their out-degree is non-zero, which is a contradiction. Therefore, there must be at most one.

   (c) The algorithm traverses each vertex, but it starts at the point it left off on the previous vertex, so there is only one walk across the matrix.

2. The entries on the diagonal are how many edges enter or leave the vertex. The other entries are equal to the additive inverse of the number of edges that go from vertex i to j or from j to i. It's also symmetric, and the rows and columns add to zero although it's not important.

3. Diameter(V)

```
Data: v : vertex
Result: diameter and height of tree
if $v = null$ then return $0, -1$;
lDiam, lHeight = Diameter(left(v));
rDiam, rHeight = Diameter(right(v));
height = Max(lHeight, rHeight) + 1;
return Max(lHeight + rHeight + 2, lDiam, rDiam), height;
```

This algorithm is $Theata(V)$, since every vertex is only visited once.