HW9 Theory

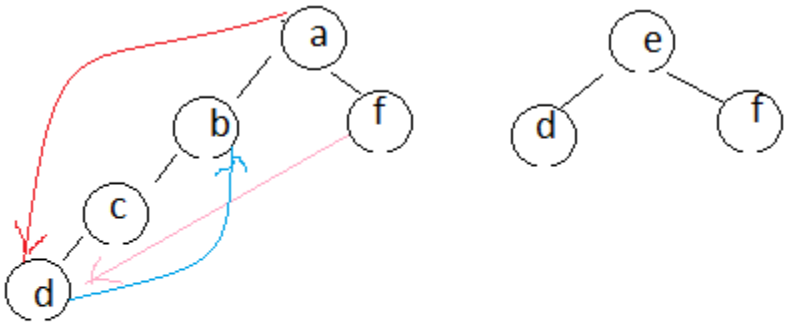
Alexander Kazantsev

March 9, 2016

Problem 6.2

If each T_i is a vertex, and the direction of the graph points to each T_i that needs completion before T_j DFS can be leveraged. First DFS on each vertex should be computed, with each set generated off of each vertex used as a reference for every other vertex referencing that set. This will generate at minimum one topologically ordered set, or at maximum T_n sets. Each set should be walked backwards, with their times t_i summed.

Problem 6.8



Problem 6.10

```
CountDepth( G, r )
    label r as discovered
    count = 1
    for all edges from v to w in G.adjacentEdges( r )
```

```
            if w not discovered
                count += CountDepth( G, w)
        return count
IsRooted ( G )
        for all v in G.vertices
            result = CountDepth( G, v )
            if result == length( G.verticies )
                return true
        return false
```

Problem 6.14

under the assumption that the paths are unweighted

Time complexity is $v^2 + v * e$

```
Paths( G, v,paths = [ ] )
        result = [ ]
        for e in G[v].edges
            result.append( Paths(G, e, paths + [ v ] ) )
        return result
LongestPath( G )
        result = [ ]
        for v in G.vertices
            result.append( Paths( G, v ) )
        return max( result )
```

Problem 6.15

(f,d,b,c)

Problem 6.20

too late to write this algo, but it should utilize DFS

Problem 7.3

Part c