CS 381
HW 6
Tian Qiu

1.

```
Relax(u, v, w)
        If v.d > u.d + w(u, v)
                v.d = u.d + w(u, v)
                v.pi = u

Initialize-Single-Source(G, S)
        For each vertex v in G.V
                v.d = infinity
                v.pi = nil
        s.d = 0

DAG-Shortest (G, W, S)
        Topologically sort Vertices of G                              // O(V + E)
        Initialize-Single-Source(G, S)                                // O(V)
        For each vertex u taken in topologically sorted order         // O(V + E)
                For each vertex v in G.adj[u]
                        Relax(u, v, w)                                // O (1)
```

Ford-Fulkerson (G, s, t)

For each edge (u,v) in G.E

(u,v).f = 0

// replace the condition for the while loop after this line

while True

// construct G_f

for (u, v) in G.E // O(E)

$c_f(u,v) = c(u,v) - f(u,v)$

$c_f(v,u) = f(u,v)$

// find path p

DAG-shortest(G_f, c_f, s) // O(V + E)

If there does not exist a path p from s to t in the residual network G_f

Break // no more p

Else

// replace the while loop before this line

$c_f(p) = min\{c_f(u,v)$: (u,v) is in p}

for each edge (u, v) in p

if (u, v) in E

(u, v).f += $c_f(p)$

else

(v, u).f -= $c_f(p)$

Time analysis:

Since requirement for maximum flow networks are

1. No self-loop
2. Every other vertex will be at least on path from s to t

So, V <= E – 1

O(V) = O(E)

Construction G_f: O(E)

Find shortest path p: O(V + E)

O (V + E + E) = O (3*E) = O (E)

So, the while loop is O(E)

2.

We assume this a flow network. So

1. No self-loop
2. Every other vertex will be at least on path from s to t

Idea:

Set all capacity of directed edges as one.

Then max flow is exactly the same as min cutting, also it is the number of edges that we have to fail so that there would be no paths at all between her node and her friend.

1. Use Edmond-Karp algorithm as described above to find the maximum flow. // O(VE^2)

2. Get max flow number |f|, and it is the answer. // O(1)

Time analysis:

The time for Ford-Fulkerson algorithm is O(VE^2)