The Ford-Fulkerson Algorithm

Now we are prepared to turn the Ford-Fulkerson method into a version of the Ford-Fulkerson algorithm. In this algorithm, we will look for some augmenting path and use that to increase the flow in the network by the amount of the path's residual capacity cf. We continue to do this until no such augmenting path exists. The following is pseudo-code for the basic Ford-Fulkerson algorithm:

Algorithm 1 Ford Fulkerson Algorithm

```
FordFulkerson(G, s, t)
   for each edge (u, v) ∈ E[G] do
      f(u, v) ← f(v, u) ← 0
   while there exists a path p from s to t in G_f do
      c_f(p) ← min{c_f(u, v), |(u, v) ∈ p}
      for each edge (u, v) ∈ p do
         f(u, v) ← f(u, v) + c_f(p)
         f(v, u) ← f(u, v)
```

In analyzing the performance of this algorithm, we see that the algorithm's complexity depends upon how we determine the augmenting path p in the test of the while loop. Without loss of generality, let's assume that all of the capacities are integers. Also, let $\hat{f}$ denote the maximum flow of the network.

Theorem: The basic Ford-Fulkerson algorithm runs in time $\Theta(|E| \cdot |\hat{f}|)$.

Proof: We can see from the pseudo-code that the initialization loop requires $\Theta(|E|)$ time. The main while loop runs at most $|\hat{f}|$ times since the flow value must increase by at least one unit per iteration (thus the assumption of integral capacities). Since the main while loop includes an update in the flow along each edge of the augmenting path p, and p contains $O(|E|)$ edges, the total complexity for the algorithm must be $\Theta(|E| \cdot |\hat{f}|)$. QED