Chapter 6. Section 2

Algorithm Ford_Fulkerson

```
INPUT:   st-network (N, s, t), capacity function c, flow function f
OUTPUT:  maximal f-unsaturated tree in N and augmented flow function f
  max:=False;
  while not Max do
      T:=Maximal_Unsaturated_Tree(N, s, t, c, f)   # Maximal_Unsaturated_Tree is specified below
      if t ∈ T then
          find the unique path P from s to t in T
          f:=Augment_Flow(N, s, t, f, c, P)   # Augment_Flow is specified below
      else max:=True
      fi
  od
  return(f, T)   # Note: the set of edges leaving T are a minimal st-cut for N
```

The heart of Ford-Fulkerson is an adaptation of the Maximal_Rooted_Network_Subtree algorithm from Week 5, Lecture 1:

Algorithm Maximal_Unsaturated_Tree

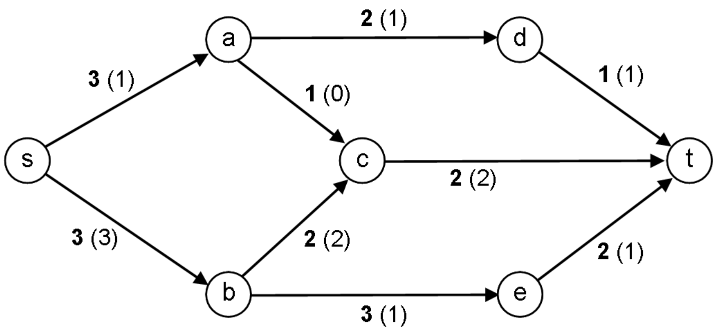
```
INPUT:   st-network (N, s, t), capacity function c, flow function f
OUTPUT:  maximal tree T in N in which all paths from s are f-unsaturated
  T := {s};
  max:=False;
  while not Max and t ∉ T do
      if there is a forward edge e leaving T with f(e) < c(e) then
          add e to T
      elif there is a backward edge e leaving T with f(e) > 0 then
          add e to T
      else max:=True
      fi
  od
  return(T)
```

Algorithm Augment_Flow

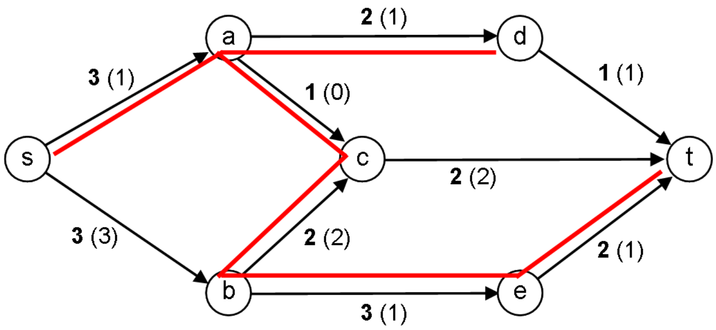
```
INPUT:   st-network (N, s, t),capacity function c, flow function f, f-unsaturated st-path P in N
OUTPUT:  augmented flow f
  sl(P) := 9999;   # Slack value for P: initialised to be higher than max possible flow value
  for e ∈ P do
      if e is a forward edge in P then sl(e) := c(e) − f(e) else sl(e) = f(e) fi
      if sl(e) < sl(P) then sl(P) := sl(e) fi
  od
  for e ∈ P do
      if e is a forward edge in P then f(e) := f(e) + sl(P) else f(e) = f(e) − sl(P) fi
  od
  return(f)
```

Example

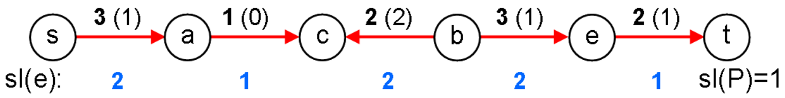
In the network shown here, the capacity $c(e)$ on edge e is shown as in bold and the flow value $f(e)$ is given in parenthesis. So edge e is labelled $\boldsymbol{c(e)}$ $(f(e))$.



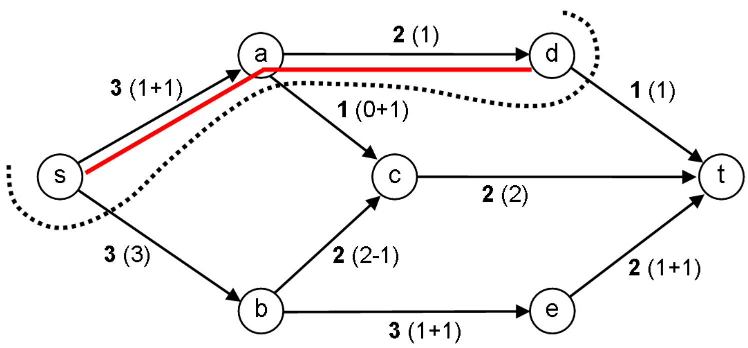
The Ford-Fulkerson algorithm applies Maximal_Unsaturated_Tree. The result is shown in heavy red edges:



Since the tree contains vertex t an f-unsaturated st-path P may be found. Note that the edge from c to b is a *backward* edge:



The figures in blue beneath each edge are the slack values found by Augment_Flow. The minimum value is sl(P) and this is used to augment the flow function f as shown below:



When Maximal_Unsaturated_Tree is applied again, the tree T does *not* contain t. So no further augmentation of f is possible and the Ford-Fulkerson algorithm terminates, returning f and the tree T:

- f is a *maximum st-flow*. Its value is 5: the sum of the f values leaving s which is equal to the sum of the f values entering t.
- T identifies a *minimum st-cut*, indicated by the dotted line: the edges directed out of T separate s from t. The sum of the f values on these edges, $3 + 1 + 1 = 5$, is the value of the minimum cut and is equal to the maximum flow value.