

# ECS 122A Homework 4

Hardy Jones  
999397426  
Professor Gysel  
Fall 2014

1. We can find a possible augmenting path if we modify both of the given algorithms.

For DFS, we do not need to look at each vertex when starting the algorithm. We already know we want to find a path from  $s$  to  $t$ , so we just need to check for paths from  $s$ .

For DFS-VISIT, we should check if the current vertex is  $t$ . If it is, then we've constructed an  $st$ -path, and we can just return it. Otherwise, we continue as in the original algorithm. Each time we visit a new vertex, we add it to the end of the current path. If what was returned from our visitation is not an empty list, then we have an  $st$ -path, so we can return it. Otherwise, we continue searching for paths. If at the end of the search, we didn't find  $t$ , we return the empty list.

---

```
function AUGMENTING-PATH( $G_f$ )  
  for each  $v \in G_f$  do  
     $v$ .visited  $\leftarrow$  FALSE  
  end for  
  return ST-PATH( $G_f$ ,  $G_f.s$ , APPEND(NIL,  $G_f.s$ ))  
end function
```

---

---

```
function ST-PATH( $G$ ,  $v$ ,  $l$ )  
   $v$ .visited  $\leftarrow$  TRUE  
  if  $v == G.t$  then  
    return  $l$   
  end if  
  for each  $u \in G.Adj(v)$  do  
    if  $u$ .visited == FALSE then  
       $path? \leftarrow$  ST-PATH( $G$ ,  $u$ , APPEND( $l$ ,  $u$ ))  
      if  $path? \neq$  NIL then  
        return  $path?$   
      end if  
    end if  
  end for  
  return NIL  
end function
```

---

2.

3. (a) True

$$v(f) = f_{OUT}(s) - f_{IN}(s) = f_{OUT}(s) = f_{IN}(u_1) = f_{OUT}(u_1) = f_{IN}(u_2) = \cdots = f_{IN}(t)$$

(b) False Minimum cuts are equivalent to maximum flow by the Max-flow min-cut theorem.

(c) False The while loop in FORD-FULKERSON has complexity  $O(|E|)$  since the path can be found with either BFS or DFS, both of which have complexity  $O(|E|)$ .

(d) True This satisfies both the capacity and the conservation constraints.