Student who wrote the solution to this question: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai
Students who read this solution to verify its clarity and correctness: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai

In our algorithm, we modify the original graph G to new graph G'. In the original graph, for each node v(except s and t), if v has more than one income paths and more than one outcome paths, then we modify it. For such v, we add a new node v', a new edge (v, v'), move all outcome path of v to v'. In a other word, v' inherited all outcome paths from v, and v lose all original outcome paths and have only one outcome path (v,v'). Then we do Edge-Disjoint-Paths algorithm in lecture slide on new graph G' and get max set of path. Finally, for each path in the set, return to it's original shape in G, and return the set.

Pseudocode:

```
Node-Disjoint-Paths (G, s, t):
    G' = Modify(G, s, t)
    P = Edge-Disjoint-Paths(G', s, t) (exactly the same as in lecture slide 14, page 20 ~ 21)
    P' = Restore(P)
    return P'

Modify(G=(V,E), s, t)       O(nm)
    G' = G
    for all node v not s or t in V':    O(n)
        in = 0
        out = 0
        outset = {}
        loop through all edge e in E':      O(m)
            if e is income edge to v:
                in ++
            if e is outcome edge to v:
                out ++
                outset add e
        if in > 1 and out > 1:
            V' add v'
            E' add (v, v')
            for all edges in outset, set the start vertex to v'
         return G'

Restore(P)
    P' = P
    for all path p in P':
        loop through p, for all edge looks like a -> v -> v' -> b, reduce it to a -> v -> b
        (key is to find node who has apostrophe in the name)
    return P'
```

Time complexity: O(nm)
Modify takes O(nm) time, reason illustrated in pseudocode. After Modify the number of nodes and edges are less than the double of origin. Edge-Disjoint-Paths takes O(4nm) = O(nm), reason illustrate in lecture slide. Restore takes O(2m) = O(m) because it will check each edge at most once. In total O(nm)

Correctness
We can assume Edge-Disjoint-Paths is correct. For every nodes(except s, t) who has more than one income edge and outcome edge, it can be in more than one edge-disjoint path, but only one node-disjoint path. After the modification, there is edge (v, v'), so for all path pass v, it must path (v, v'). So, there can be only one edge-disjoint path, same as node-disjoint path. For other nodes whose number of income path or outcome path is 1, the number of edge-disjoint path and node-disjoint path pass it is 1, the same.

Therefore the number of elements in max edge-disjoint path set of G' is the same as number of elements in max node-disjoint path set of G. Hence correct.