

Algorithm Program Assignment 3

b97502049 電機四 蔡昀達

1. In your report, please write a pseudo code for is_spanning_tree and analyze its complexity.

Ans.

To realize the is_spanning tree function, we need to check out the following things and judge if the file we read in satisfies the spanning tree property of the original graph we construct before.

We need to check out:

1. To check out that if the vertex number of current tree is the same as the original graph's.
2. To check out if every edge in every vertex in the current tree exist in the original graph. If the edge is exist, the weight of this edge must be the same as the one in original graph.

If all of above can be satisfied, we can claim that the current tree is a spanning tree of the original graph.

Pseudo code:

Suppose that G is the origin graph, T is the current tree. And G.v represent each vertex in the graph, also G.v.e means that all the edge of G.v.

```
if ( number of G.v != number T.v ) return false
for each u in T.v
    if ( ! (G contain the same key as u ) ) return false
    for each x in T.v.e
        if ( !G contain the same edge with the same key) return false
```

```
return true
```

Complexity analysis:

```
if ( number of G.v != number T.v ) return false       $\theta(1)$ 
for each u in T.v                                      $O(V)$ 
    if ( ! (G contain the same key as u ) ) return false  $O(V)$ 
    for each x in T.v.e                                $O(E)$ 
        if ( !G contain the same edge with the same key) return false  $O(E)$ 
return true
```

Hence, the total time complexity is

$O(V) * (O(V) + O(E) * O(E))$

$O(V) * (O(V) + O(E^2))$

The answer is $O(V^2) + O(V * E^2)$.

2. Table of reunning result.

input size	DFS			BFS			MST(prim)		
	vertices/edges	runtime(s)	memory(MB)	vertices/edges	runtime(s)	memory(MB)	vertices/edges	runtime(s)	memory(MB)
n4	1.3333333333	0.000135	3.38672	1.3333333333	6.70E-005	3.38672	1.3333333333	6.40E-005	3.38672
n10	1.1111111111	0.000118	3.38672	1.1111111111	9.20E-005	3.38672	1.1111111111	0.000112	3.38672
n100	1.0101010101	0.018962	3.38672	1.0101010101	0.000537	3.38672	1.0101010101	0.001304	3.38672
n1000	1.001001001	0.00499	3.64453	1.001001001	0.004957	3.64453	1.001001001	0.068045	3.64453
n10000	1.00010001	0.05199	6.21094	1.00010001	0.052033	6.21094	1.00010001	6.20682	6.21094
n50000	1.0000200004	0.281314	18.8516	1.0000200004	0.372306	18.8555	1.0000200004	163.812	19.0469

Plot the result:

