97 中央大學 資料結構與演算法

1.

```
if    Find(u)≠Find(v)
{
      add (u, v) to spanning tree
      Union(u, v);
}
else
      會形成 cycle, reject (u, v);
```

2.

ABCDE

3.

(a) SWAP(root->left);
(b) SWAP(root->right);
(c) p = root->left;
(d) root->left = root->right;
(e) root->right = p;

4.

(a) 對於 G 中的每個點 v, 皆有 path 可以連到 G 中的其他不等於 v 的點

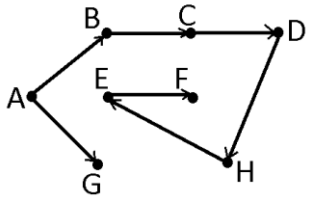
(b)

iteration	S	Vertex Selected	A	B	C	D	E	F	G	H
1	A	A	0	1	∞	∞	∞	∞	1	∞
2	A, B	B	0	1	5	∞	2	∞	1	∞
3	A, B, G	G	0	1	5	∞	2	∞	1	2
4	A, B, G, E	E	0	1	5	∞	2	3	1	2
5	A, B, G, E, H	H	0	1	5	∞	2	3	1	2
6	A, B, G, E, H, F	F	0	1	4	∞	2	3	1	2
7	A, B, G, E, H, F, C	C	0	1	4	5	2	3	1	2
8	A, B, G, E, H, F, C, D	D	0	1	4	5	2	3	1	2

➔ the shortest path: A->B->E->F->C->D

(c)

作 DFS 後的 graph 如下



Back edge : $\overline{FC}$, $\overline{FH}$, $\overline{EA}$

Forward edge: $\overline{BE}$

Cross edge: $\overline{GE}$, $\overline{GH}$

5.

```
int BSOX( S: string )
{
    count <- 0;
    L <- 1;
    U <- length(S);
    while( L ≦ U )
    {
        m <- ( L+U )/2;
        if( S[m] == 'O' )
        {
            count <- m-L+1;
            L <- m+1;
        }
    }
}
```

主程式如下

```
int NumOfO( A:array )
{
    C <- 0;
    for ( i<-0 to n )
        C <- C + BSOX( A[i][] );
    return C;
}
```

6. ?

7.

(a) False

if the heaviest edge is a bridge, then it must in the Minimum spanning tree

(b) True

若有一個 MST 含此 e, 即令含 e 的 cycle 為 C

則此 MST 必不含 C 中之每一個 edge

令 G 中其他的 edge f 不屬於 C

則將 f 加入 MST 中，形成一個 cycle，去除掉 cycle 中其他之一 edge

形成 weight 較小的 MST

(c) True

證法類似(b)

(d) True