

# Pseudocode:

**Graph:**  $O(V)$

Add each V into a Graph list

**addEdge:**  $O(E)$

Add each E into a Graph vector

**DFSUtil:**  $O(E)$

visited[v] = true;

For i = edge 0 to edge n-1 :

If not visited :

DFSUtil(i, visited);

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

visited[0 to V-1] = false;

DFSUtil(v, visited);

**counter:**  $O(V)$

//回傳從某個點出發可以走到幾個沒被走過的點 用來找出可以走到最多的 然後連到他且走過的就不能走了

for i = 0 to V-1

if visited[i]==1 && baseline[i]==0 do count++;

}

return count;

**find\_nearest:**  $O(V)$

for i = 0 to V-1:

if visited[i]==1:

calculate the distance of (i to endpoint)

if distance < temp\_short:

shortest\_distance = distance;

shortest\_startpoint = i;

return shortest\_distance, shortest\_startpoint;

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

load data;  $O(V+E)$

while all\_connected == 0:  $O(V*(V+E)+V*V) = O(V^2+V*E)$

do DFS(0); //從開始進行 DFS  $O(V+E)$

for i = 0 to V:  $O(V)$

if visited[i]==0:

if counter(DFS(i)) > temp\_max:

```

        temp_max = counter(DFS(i));
        max = i;
    }
}
}
if max != 0:
    data = find_nearest(max);
    Graph.addEdge(max);
}
else if (max == 0) all_connected = 1;
}

```

## Time complexity analysis:

Assume:

We have **V** airport and **E** flight.

**Time complexity =  $O(V+E) + O(V*(V+E)+V*V) = O(V^2+V*E)$**

Experimental results:

Input:

1 8

2 10 20

3 16 18

4 3000 1000

5 3542 1111

6 654 321

7 951 753

8 5 6

9 0 0

10 1 0

11 0 6

12 6 2

13 6 5

14 5 0

15 2 4

16 3 4

17 4 2

18 7 2

Output:

1 3

2 0 1

3 6 7

4 2 3

Input:

10

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

0 1

1 2

2 4

4 5

5 6

6 7

7 9

8 3

Output:

1

7 8

Case1 output:

53

2237 143

223 31

4496 70

661 4529

4337 319

898 34

3983 3250

457 220

4238 8

513 18

1887 184

4688 124

3488 24

1720 139

3085 45

1112 66

3874 32

3191 65

4518 183

939 78

4662 2628

3872 58

4008 1195

504 11

858 99

Case3 output:

305

1018 59

235 600

180 972

306 567

997 594

910 225

898 2

119 460

191 185

397 141

808 875

776 201

233 434

918 129

746 171

459 400

1007 274

714 516

105 143

537 190

986 796

238 832

429 162

133 602

429 330

148 619

523 451

525 206

170 566

397 476

655 232

870 295

32 310