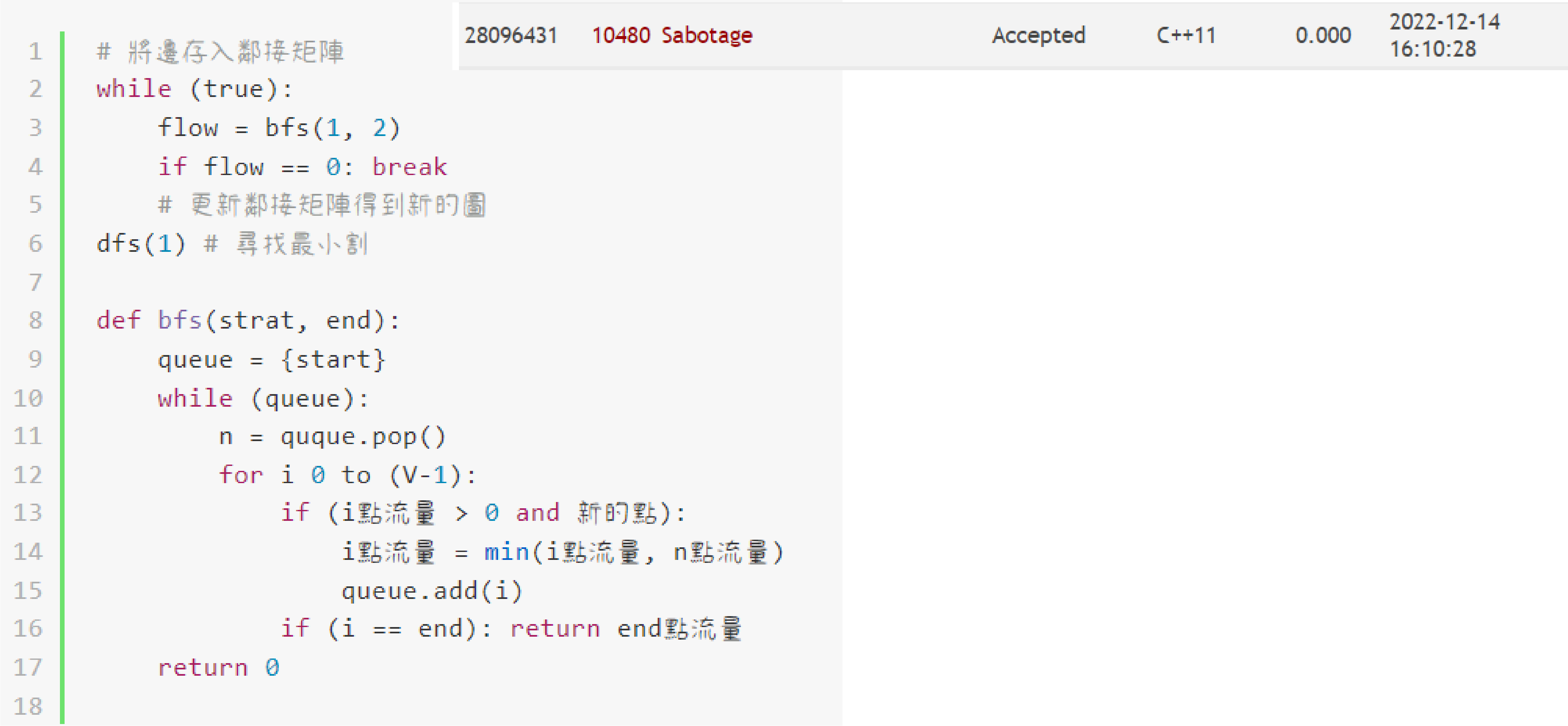
專題4

110502518陳文獻

方法1 Flow (Edmonds–Karp)



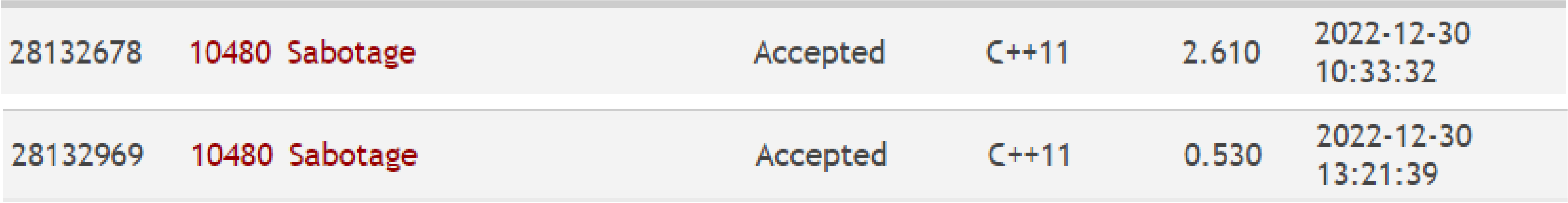
28096431 10480 Sabotage Accepted C++11 0.000 2022-12-14 16:10:28

```
# 將邊存入鄰接矩陣
while (true):
    flow = bfs(1, 2)
    if flow == 0: break
    # 更新鄰接矩陣得到新的圖
dfs(1) # 尋找最小割

def bfs(strat, end):
    queue = {start}
    while (queue):
        n = quque.pop()
        for i 0 to (V-1):
            if (i點流量 > 0 and 新的點):
                i點流量 = min(i點流量, n點流量)
                queue.add(i)
            if (i == end): return end點流量
    return 0

```

方法2 隨機合併點(Karger’s algorithm)

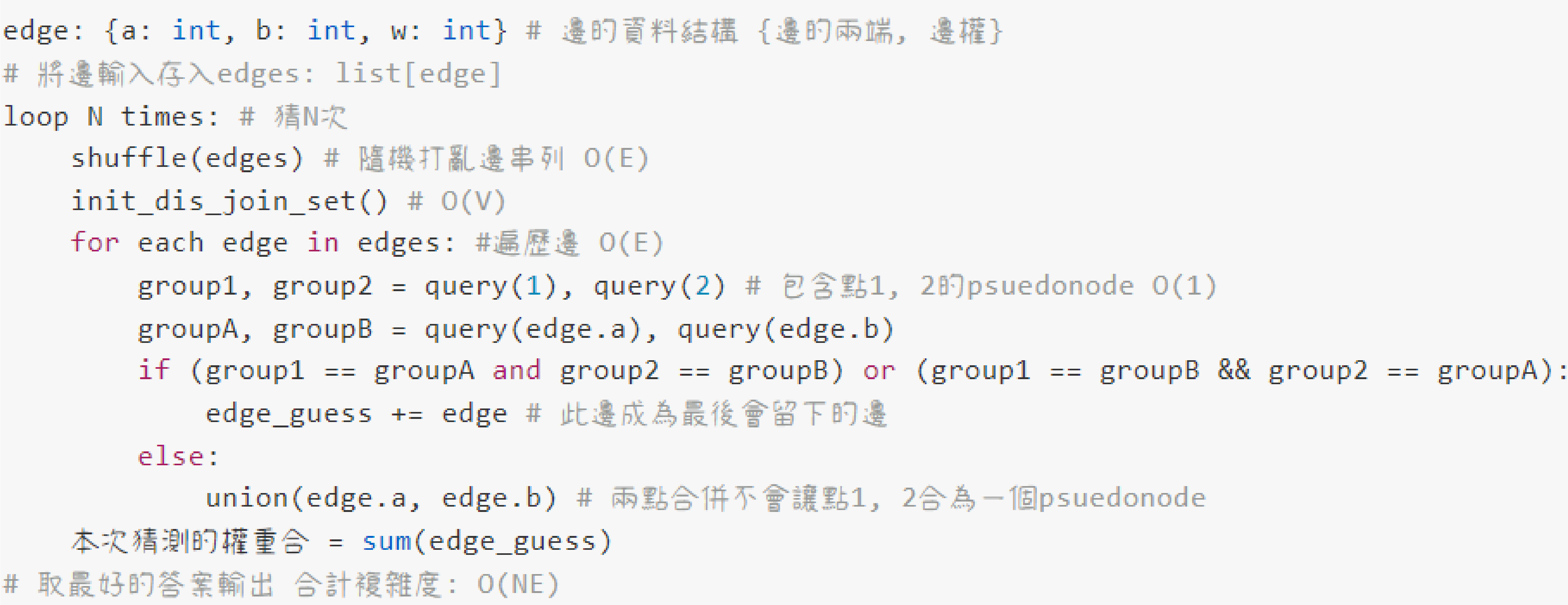
- 使用並查集維護pseudonode
- 每次打亂儲存邊的串列
- 若合併該邊的兩點會使點1, 2成為同個node -> 跳過並記錄此邊
- 若不會 -> 合併兩點pseudonode

UVA結果: N(隨機合併的次數)設定50000/10000



28132678	10480 Sabotage	Accepted	C++11	2.610	2022-12-30 10:33:32
28132969	10480 Sabotage	Accepted	C++11	0.530	2022-12-30 13:21:39

Pseudocode



```
edge: {a: int, b: int, w: int} # 邊的資料結構 {邊的兩端, 邊權}
# 將邊輸入存入edges: list[edge]
loop N times: # 猜N次
    shuffle(edges) # 隨機打亂邊串列 O(E)
    init_dis_join_set() # O(V)
    for each edge in edges: #遍歷邊 O(E)
        group1, group2 = query(1), query(2) # 包含點1, 2的psuedonode O(1)
        groupA, groupB = query(edge.a), query(edge.b)
        if (group1 == groupA and group2 == groupB) or (group1 == groupB && group2 == groupA):
            edge_guess += edge # 此邊成為最後會留下的邊
        else:
            union(edge.a, edge.b) # 兩點合併不會讓點1, 2合為一個psuedonode
    本次猜測的權重合 = sum(edge_guess)
# 取最好的答案輸出 合計複雜度: O(NE)
```

資料結構設計說明(disjoin set)

```
query(x)
    if (djs[x] == x)
        return x
    djs[x] = query(djs[x])
    return djs[x]
```

(with “path compression”)
O(1)

```
Union(a0, b0, w)
    a = query(a0); b = query(b0)
    if (a == b) return
if (rank[a] < rank[b]) djs[a] = b
    else if (rank[a] > rank[b]) djs[b] = a
    else djs[a] = b, ++rank[b]
```

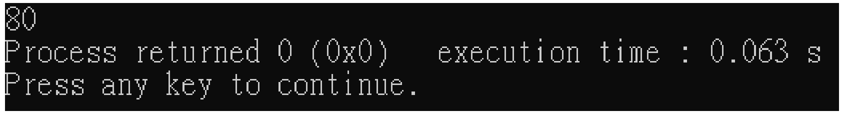
(with “merge by rank”)
O(1)

實驗結果(方法2的正確率、時間):

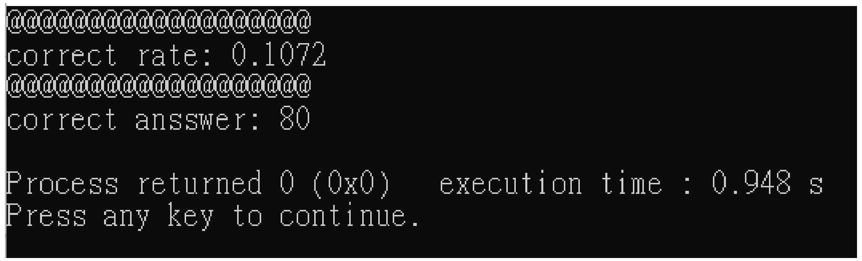
flow

random(取樣10000次)

V=5, E=8(uva範例)

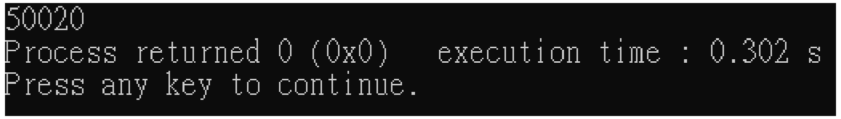


```
80
Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

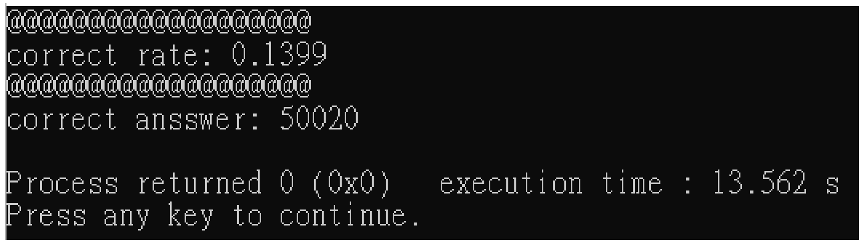


```
@@@@@@@@@@@@@@@@@@@
correct rate: 0.1072
@@@@@@@@@@@@@@@@@@@
correct ansswer: 80

Process returned 0 (0x0)   execution time : 0.948 s
Press any key to continue.
```

V=1000, E=10000 (稀疏圖)

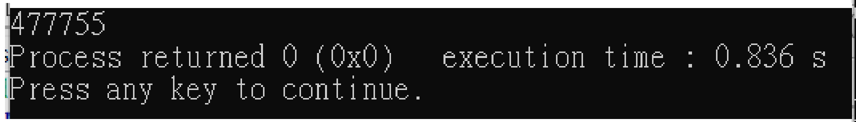


```
50020
Process returned 0 (0x0)   execution time : 0.302 s
Press any key to continue.
```

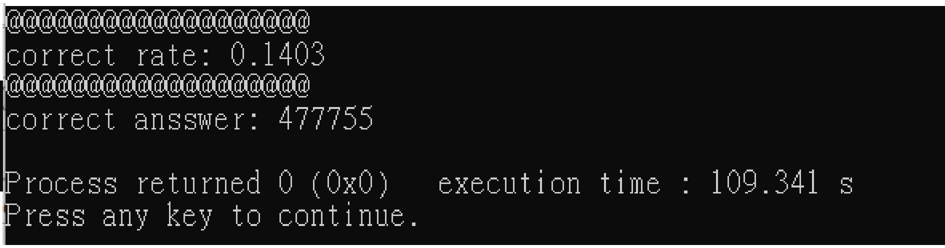


```
@@@@@@@@@@@@@@@@@@@
correct rate: 0.1399
@@@@@@@@@@@@@@@@@@@
correct ansswer: 50020

Process returned 0 (0x0)   execution time : 13.562 s
Press any key to continue.
```

V=1000, E=100000 (稠密圖)



```
477755
Process returned 0 (0x0)   execution time : 0.836 s
Press any key to continue.
```



```
@@@@@@@@@@@@@@@@@@@
correct rate: 0.1403
@@@@@@@@@@@@@@@@@@@
correct ansswer: 477755

Process returned 0 (0x0)   execution time : 109.341 s
Press any key to continue.
```