Programming Assignment #3
Cycle Breaking

B11901164 Bing-Wei, Chen

June 15, 2024

1 Undirected Graph

For an undirected graph (Figure 1), store the edges using the *addEdge* function into a vector E of type *Edge* (Figure 2). Then, run the *mstKruskal* function (Maximum Spanning Tree by Kruskal's Algorithm) [2] to identify the lightest edges and store them in the vector *removedEdges* (Figure 3). Finally, sum the total weight of the removed edges and print all the removed edges (Figure 4).

```
Sample Input 1
u
8
9
0 1 3
0 2 5
1 3 10
1 4 8
2 5 9
3 4 5
3 5 11
3 6 12
4 7 6
0
```

Figure 1: sample input 1

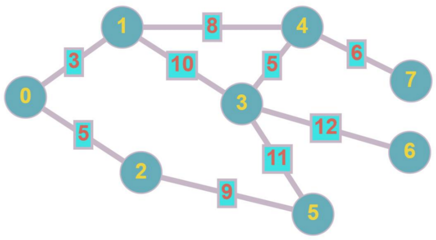


Figure 2: undirected graph

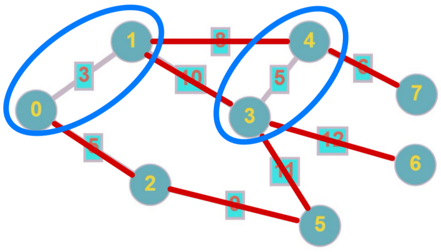


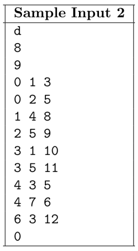
Figure 3: acyclic undirected graph

```
Sample Output 1
8
0 1 3
3 4 5
```

Figure 4: sample output 1

2 Directed Graph

For a directed graph (Figure 5), store the edges using the *addEdge* function into an adjacency list and a vector E of type *Edge* (Figure 6). Next, run the *breakDirectedGraphCycle* function to treat the graph as undirected and execute *mstKruskal* [2]. Store the removed edges in the *removedEdges* vector and erase these edges from the original graph (Figure 7). Then, iteratively push these edges back into the original graph and run *DFS* [1] to check for cycles. Remove the edge from the original graph if a cycle is detected and add it back to *removedEdges* (Figure 8 and 9). Finally, sum the total weight of the removed edges to ensure the input graph is acyclic and print all the removed edges (Figure 10).



```
Sample Input 2
d
8
9
0 1 3
0 2 5
1 4 8
2 5 9
3 1 10
3 5 11
4 3 5
4 7 6
6 3 12
0
```

Figure 5: sample input 2

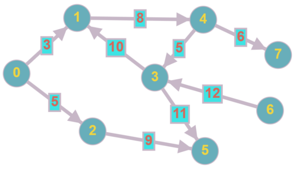


Figure 6: directed graph

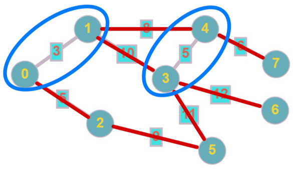


Figure 7: treat directed graph as undirected graph

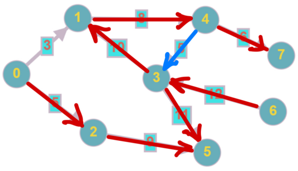


Figure 8: a cycle forms when adding edge (4, 3)

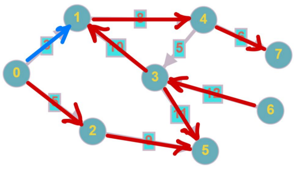
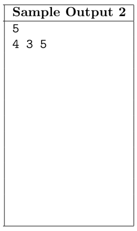


Figure 9: no cycle forms when adding edge (0, 1)



```
Sample Output 2
5
4 3 5
```

Figure 10: sample output 2

3 Statistics

Run the program on the EDA Union server.

public case	0	1	2	3	4	7	8
weight sum	5	21	-3330	-21468	0	-10515	-70938
time(ms)	0.172	0.171	0.258	1.542	0.132	0.881	11.154

Table 1: result

References

[1] Iris Hui-Ru Jiang. UNIT 6 GRAPHS PART I: Basics and BFS/DFS, Spring 2024. https://cool.ntu.edu.tw/courses/36320/files/folder/Slides?preview=5571087.

[2] Iris Hui-Ru Jiang. UNIT 6 GRAPHS PART II: Minimum Spanning Trees, Spring 2024. https://cool.ntu.edu.tw/courses/36320/files/folder/Slides?preview=5571086.