1 Minimal Spanning Tree Algorithm

1.1 Introduction

This project mainly contains

- Prim Algorithm
- Kruskal Algorithm

1.2 Kruskal Algorithm

1.2.1 Pseudo Code

Algorithm 1 Kruskal algorithm

1: **function** KRUSKAL(V, E) ▷ V denotes vertices, E denotes edges

Require: A weighted, connected map which vertices set as V and edges set as E.

Ensure: Using map G_{new} which describe the minimal spanning tree.

2: $G_{new} \leftarrow \{v_0, e_0 \mid v_0 = V, e_0 \in \emptyset\}$ ▷ v_0 has the same vertices number as V, e_0 denotes empty set

3: $E_s \leftarrow sortFromSmallToLarge\,(E)$

4: $V_{connected} \leftarrow \{v_0, v_1 \mid \langle v_0, v_1\rangle \in E_s\,[0]\}$

5: **for all** $e_i \in E_s$ **do** ▷ From small to large

6: **if** $\forall v_t \in V, v_t \in G_{new}$ **then**

7: break

8: **end if**

9: **if** $v_0 \in V_{connect}$ and $v_1 \notin V_{connect}$ s.t. $\langle v_0, v_1\rangle \in e_i$ **then**

10: add e_i to G_{new}

11: add v_1 to $V_{connected}$

12: **end if**

13: **end for**

14: **end function**

1.2.2 Flowchart

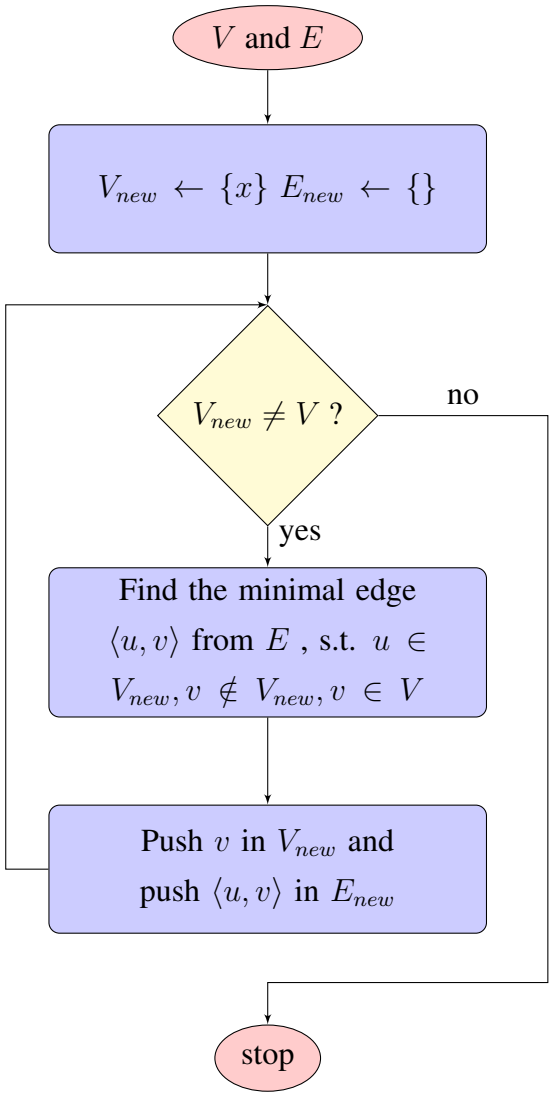

图 1: Kruskal algorithm flowchart

1.2.3 Analysis

Let v denotes the sum of vertices and e denotes the sum of edges, then, this algorithm's time complexity is: $O\left(e \log_2 e\right)$