KRUSKAL’S MINIMUM SPANNING TREE ALGORITHM

Team 6

Mishek Sambahangphe & Cesar Gonzalez

Introduction

- **Kruskal's Algorithm for Walkways:**
  - **Objective:** Implement line follower robot for tasks.
- **Benefits:**
  - **Automation:** Pick up books, transport materials, waste.
  - **Versatility:** Efficiently handles diverse tasks.
- **Cost Optimization:**
  - **Minimal Cost:** Ensures economical walkway marking.
  - **Efficiency:** Optimizes path for line follower robot.
- **Enhanced Operations:**
  - **Streamlined Logistics:** Efficient resource movement.
  - **Savings:** Time, energy, and manual effort.
  - **Reliability:** Trustworthy non-human operated vehicle.

Applications in real life

- **Restaurant Serving Robot with Double Line Sensors Following Approach [1]**

★ The serving robot is programmed to come to a specific table by mapping data.

★ Based on the line reading algorithm from two LED array line sensors implemented on microcontroller.

★ The robot follows the line marked on the floor to move to the desired table position and returns to the service counter after completing the task.

Restaurant Serving Robot

2019 IEEE International Conference on Mechatronics and Automation (ICMA)
Date of Conference: 04-07 August 2019.
Date Added to IEEE *Xplore*: 29 August 2019
INSPEC Accession Number: 18956524. **DOI:** 10.1109/ICMA.2019.8816404
Publisher: IEEE. **Conference Location:** Tianjin, China

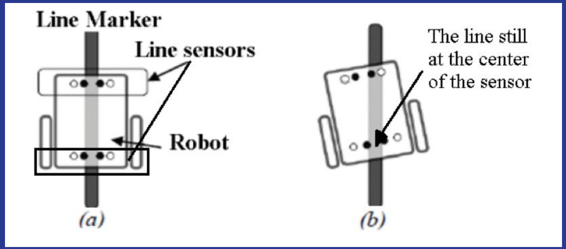


Figure 1 shows the trajectory of a line follower robot [1]

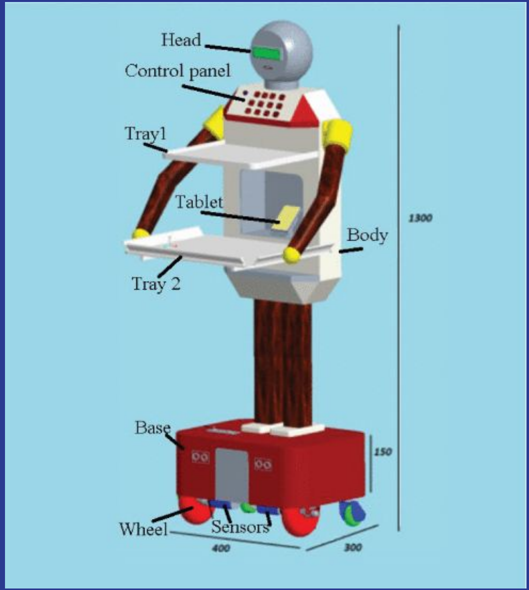


Figure 2 shows a line follower robot [1]

Sources : [1]https://ieeexplore-ieee-org.libproxy.csun.edu/document/8816404

Line follower robots & their use

- Line Follower Robot: Design and Hardware Application [2]
  - These robots may be used in various industrial and domestic applications such as to carry goods, floor cleaning, delivery services and transportation. [2]

- Design and Development of Automated Guided Vehicle with Line Follower Concept [3]
  - During the design, basic functionalities such as line follow, drive mechanism, and path planning are highlighted. It can be used as a warehouse material handling robot, and to improve the health-care system, among other things. [3]

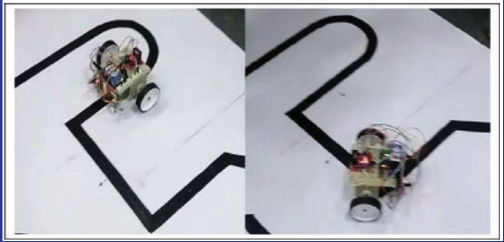


Figure 3 shows Movement of Line Follower Robot [2]

Figure 4 shows an Automated Guided Vehicle with Line Follower Concept [3]

[2] https://ieeexplore-ieee-org.libproxy.csun.edu/document/9197968
[3] https://ieeexplore-ieee-org.libproxy.csun.edu/document/10179618

How are we planning to implement our idea to the real life at CSUN campus

- We will use a computational algorithm to find a path that is cost efficient and will let us travel through all CSUN walkways

Kruskal's Minimum Spanning Tree Algorithm (MST)

- Generates a minimum spanning tree for a weighted, undirected graph.
- Sorts edges in increasing order based on the edge's weight and keeps adding nodes to the tree only if the chosen edge does not form any cycle.
- Picks the edge with a minimum cost at first and the edge with a maximum cost at last.

Sources: [4] https://ieeexplore-ieee-org.libproxy.csun.edu/document/7975216

Kruskal's Spanning Tree Algorithm [4] [5], [6], [7]

- The algorithm selects an edge with the smallest weight and checks if it is a safe edge.
- Safe edge is defined to be an edge that can be safely added to the minimum spanning tree without making a cycle or breaking its property of maintaining a minimum weight
- If the edge (u, v) is safe, the set comprising u and the set comprising v are combined to a larger set.
- If the edge is not safe, then it is rejected.
- This procedure is iterated until all edges are tested.

What is a Minimum Spanning Tree

- The Minimum Spanning Tree (MST) of a graph is the set of edges that connect every vertex contained in the original graph, such that the total weight of the edges in the tree is minimized. [4], [5], [6], [7]

Sources: [4] https://ieeexplore-ieee-org.libproxy.csun.edu/document/7975216

General applications of Kruskal's algorithms: [10], [11], [12]

- Landing cables.
- TV Network.
- Tour Operations.
- LAN Networks.
- A network of pipes for drinking water or natural gas.
- An electric grid.
- Network for roads and Rail tracks connecting all the cities.

Sources: [10] https://ieeexplore-ieee-org.libproxy.csun.edu/document/10330890
[11] https://ieeexplore-ieee-org.libproxy.csun.edu/document/1673166
[12] https://ieeexplore-ieee-org.libproxy.csun.edu/document/7154702

How Many Edges Does a Minimum Spanning Tree Have?

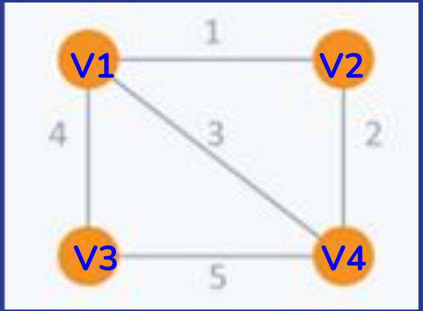
- A minimum spanning tree has precisely n-1 edges, where n is the number of vertices in the graph.

4 VERTICES IN THIS GRAPH

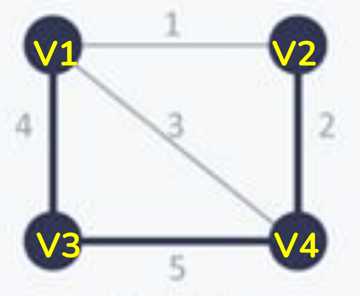
n = 4

n -1 = # EDGES, 4 - 1 = 3

3 EDGES IN THIS GRAPH FOR THE MINIMUM SPANNING TREE

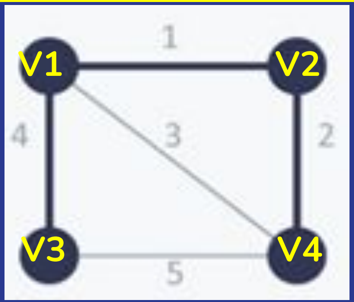


UNDIRECTED GRAPH



SPANNING TREE COST

COST = 11 = 4 + 5 + 2



MINIMUM SPANNING TREE COST

MST COST = 7 = 4 + 1 + 2

Figure 5 shows a diagram of the Kruskal algorithm
Sources: [13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

How to create a minimum spanning tree MST

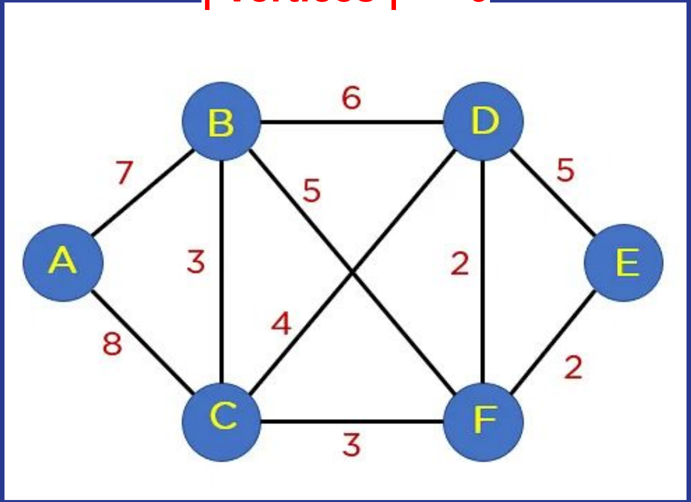
- Step 1: Sort all edges in increasing order of their edge weights.
- Step 2: Pick the smallest edge. Can have multiple edges with the same weight
- Step 3: Check if the new edge creates a cycle or loop in a spanning tree.
- Step 4: If it doesn't form the cycle, then include that edge in MST. Otherwise, discard it.
- Step 5: Repeat from step 2 until it includes all edges which are all vertices minus one.

  | Vertices | — 1 = # edges   in MST.

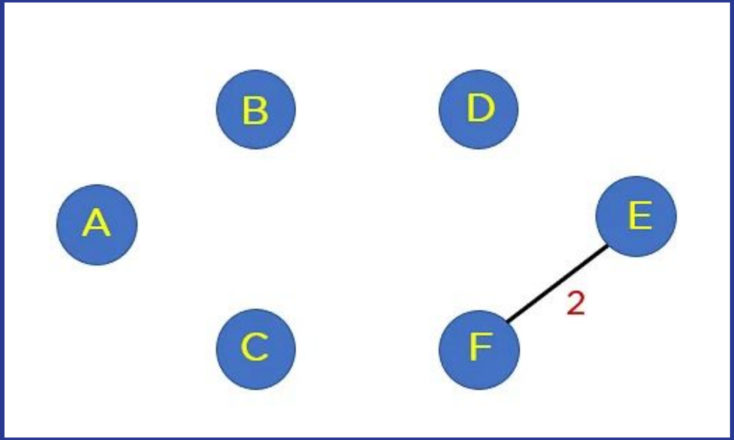
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

Example: Find minimum spanning tree MST using Kruskal's algorithm

| Vertices | = 6



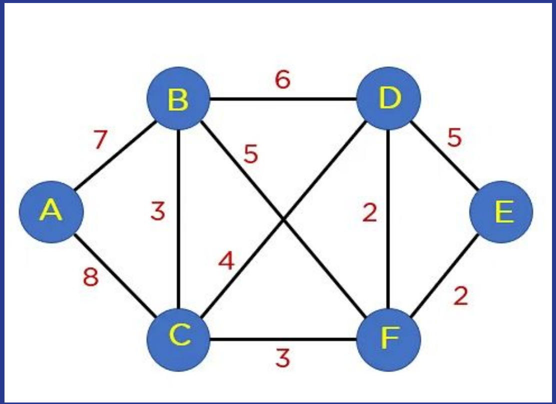
Edges	Weight
E → F	2
F → D	2
B → C	3
C → F	3
C → D	4
B → F	5
B → D	6
A → B	7
A → C	8

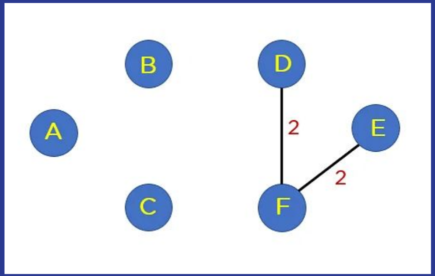


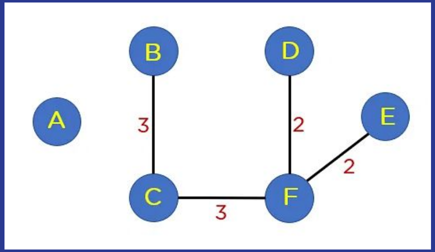
Minimum Spanning Tree Edges: | Vertices | — 1 = # edges = 5 Edges

Figure 6 shows the steps of the kruskal algorithm
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

Edges	Weight
E → F	2
F → D	2
B → C	3
C → F	3
C → D	4
B → F	5
B → D	6
A → B	7
A → C	8







Edge CD should be discarded, as it creates a loop.

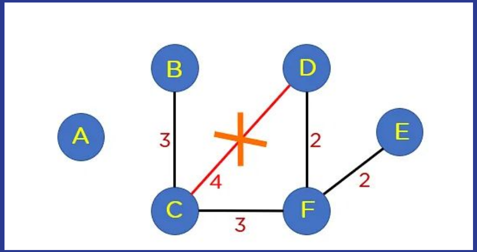
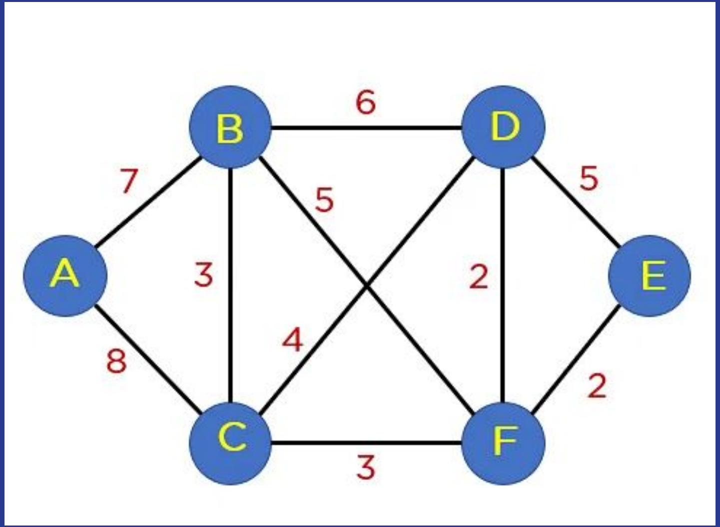


Figure 7 shows the in-between steps of the kruskal algorithm
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

Edges	Weight
E $\rightarrow$ F	2
F $\rightarrow$ D	2
B $\rightarrow$ C	3
C $\rightarrow$ F	3
C $\rightarrow$ D	4
B $\rightarrow$ F	5
B $\rightarrow$ D	6
A $\rightarrow$ B	7
A $\rightarrow$ C	8

| Vertices | = 6



Minimum Spanning Tree Edges

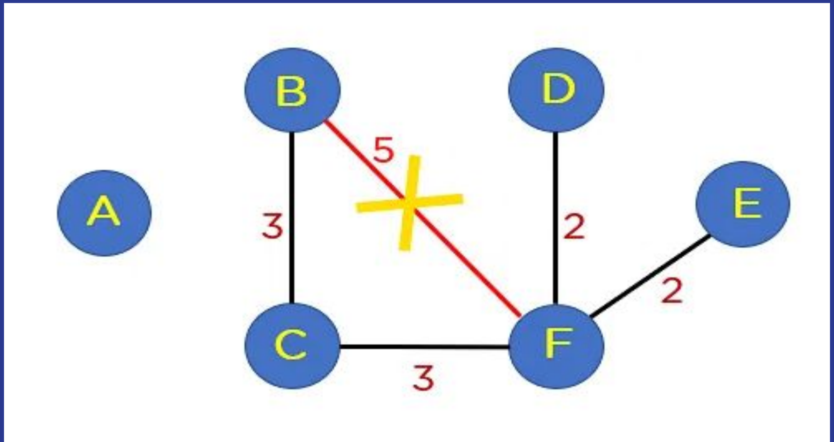
| Vertices | — 1 = # edges

5 Edges

Figure 8 shows the final steps of the kruskal algorithm
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

Edges	Weight
E → F	2
F → D	2
B → C	3
C → F	3
C → D	4
B → F	5
B → D	6
A → B	7
A → C	8

Edge BF should be discarded.



Edge BD should be discarded.

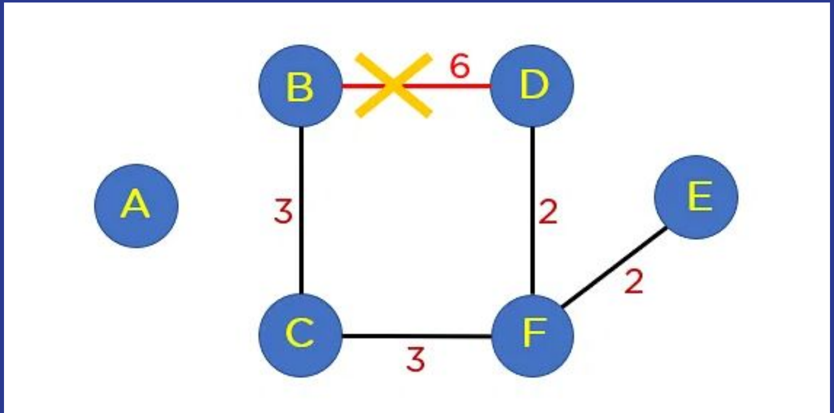
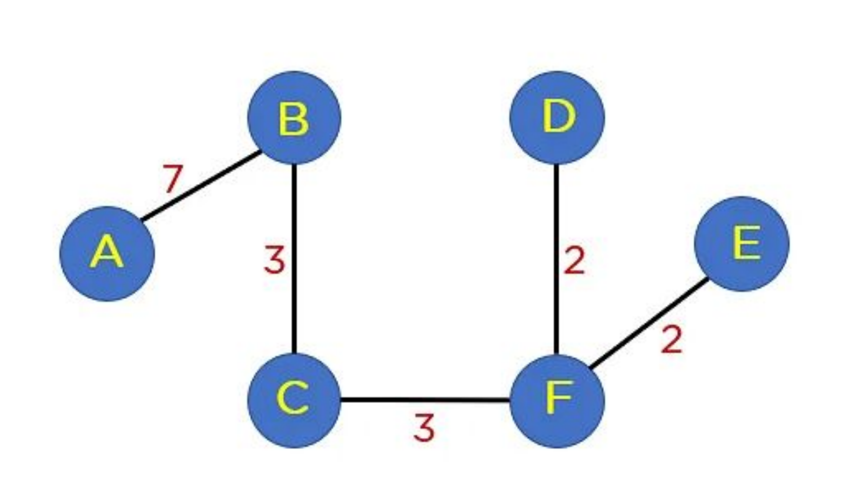


Figure 9 shows the final steps of the kruskal algorithm
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

The sum of all the edge weights in MST is equal to 17, which is the least possible edge weight for any possible spanning tree structure for this particular graph.



Minimum Spanning Tree (MST) with 5 Edges

Edges	Weight
E → F	2
F → D	2
B → C	3
C → F	3
C → D	4
B → F	5
B → D	6
A → B	7
A → C	8

Figure 10 shows the final steps of the kruskal algorithm
Sources:[13]Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10.

New tentative perspective of plan:

- Find the lowest cost possible to mark CSUN's walkways in order to use a line follower robot.

- Using Kruskal's algorithm we will find the minimal possible cost per foot length to implement this idea.

- It will be possible to use a line follower robot to perform multiple tasks like:

    - Pick up books from the Library book drops around the campus
    - Transportation of different materials, waste, goods, people (drop off people to different locations and have the robot to go back by itself)
    - Keep CSUN campus clean
    - And many more tasks.

Implementation

PORTION OF CSUN MAP [6]

Figure 11 shows a part of the CSUN campus
Source: [6]https://3dmap.csun.edu

Dataset Collection

Dataset :

We collected 48 distances that connect 37 vertices in this portion of CSUN campus

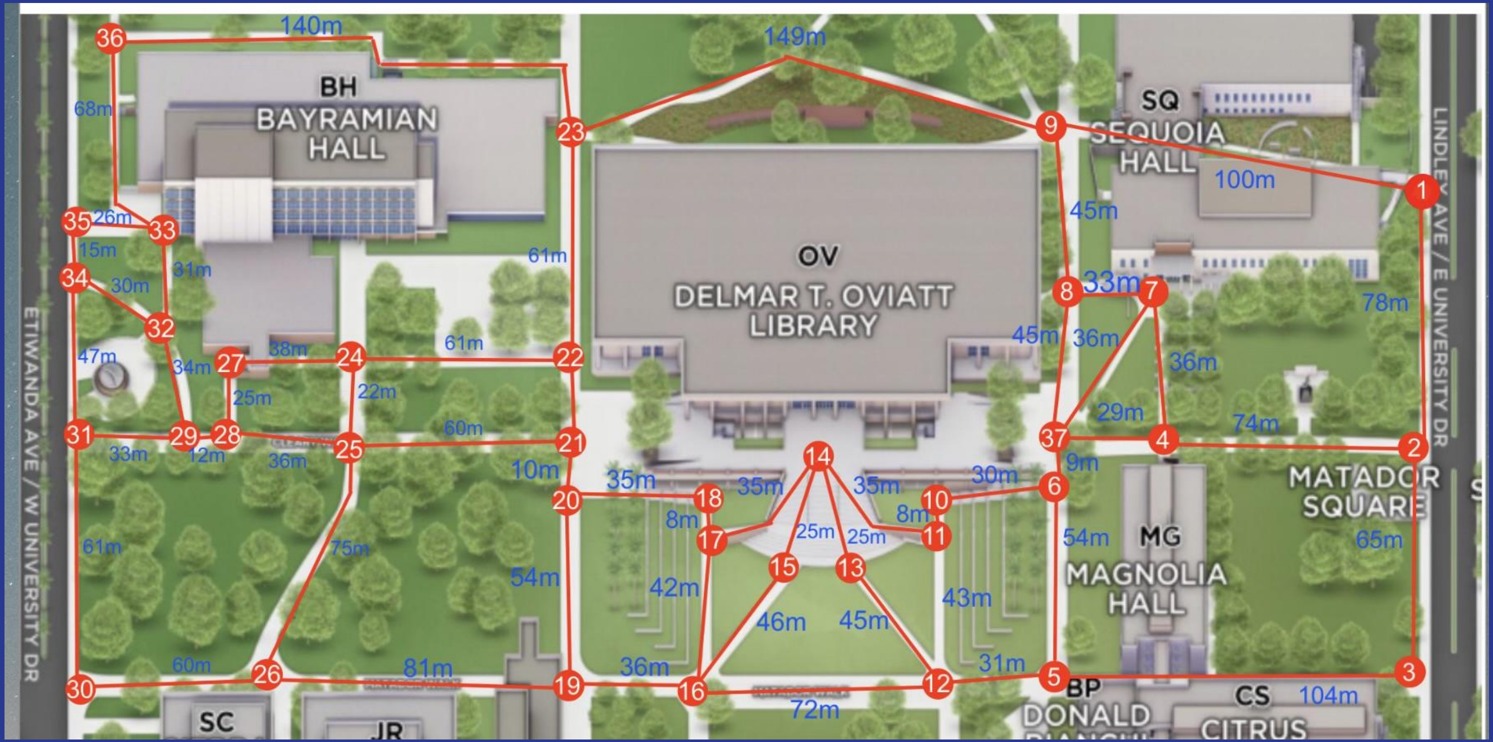


Figure 12 Map of data set collected

We used google maps to obtain the distance between each vertex.

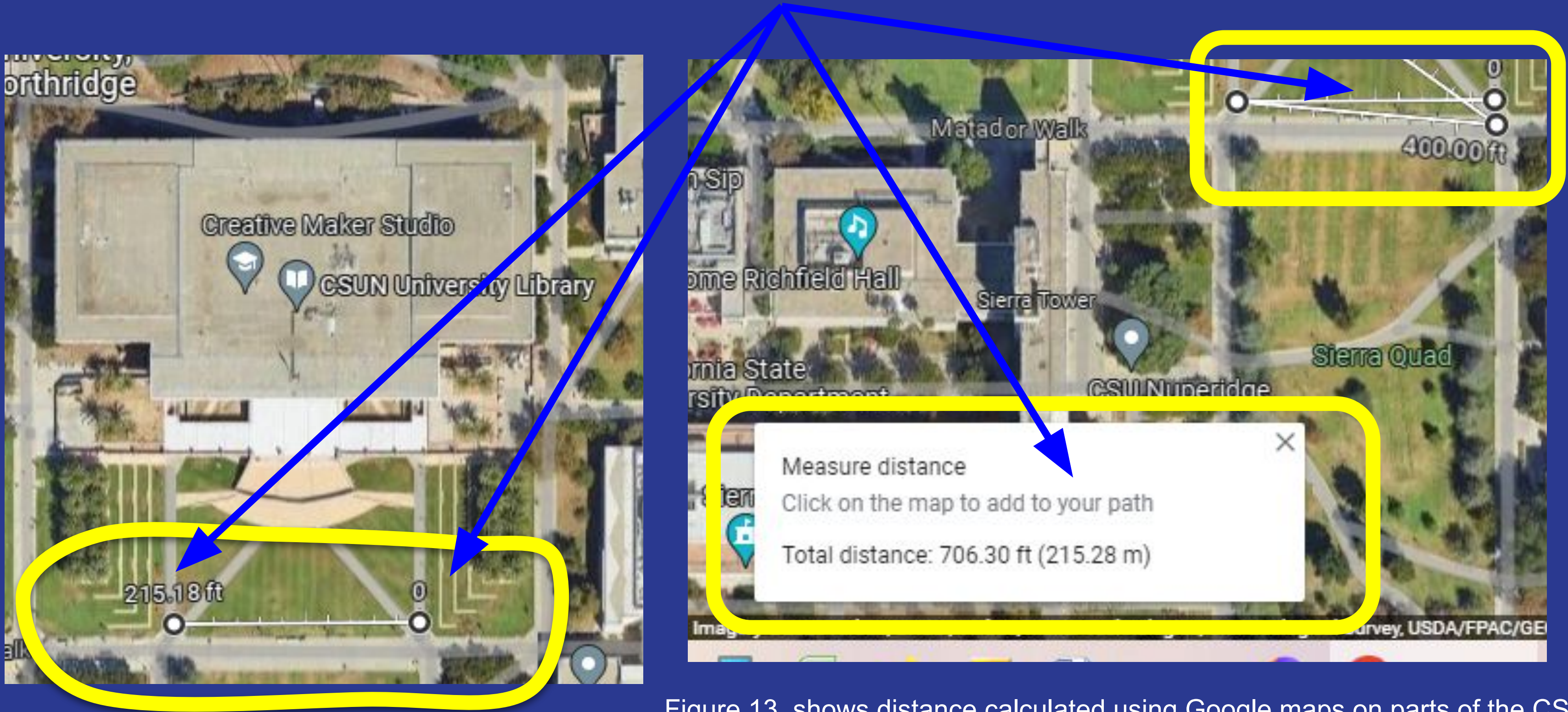


Figure 13 shows distance calculated using Google maps on parts of the CSUN campus

PSEUDOCODE

```
//List of Edges is already sorted from least to greatest
Kruskals(# of Vertices, List of Edges){

    for(# of Vertices){
        initialize all subsets with rank 0
    }
    //Loop to add Edges to the MST(Minimum Spanning Tree)
    while(# of Vertices){
        newEdge = next Edge from Edge List
        if(newEdge.Source not equal to newEdge.Destination){
            Add newEdge to Minimum Spanning Tree
            union(Source and Destination)
        }
    }
}
```

Subset class represents grouping of Vertices
Rank is the height of the Subset

Ensures no duplicate edges are added

Union Function will merge the two disjiointed trees into

Result generated by the source code

36 edges with their weight

This is the final minimum spanning tree

Following are the edges of the constructed MST:

10 -- 11 == 8

17 -- 18 == 8

6 -- 37 == 9

.

.

.

22 -- 23 == 61

30 -- 31 == 61

2 -- 3 == 65

33 -- 36 == 68

2 -- 4 == 74

1 -- 2 == 78

Total cost of MST: 1301

Results: Kruskal's Minimum Spanning Tree denoted in green lines

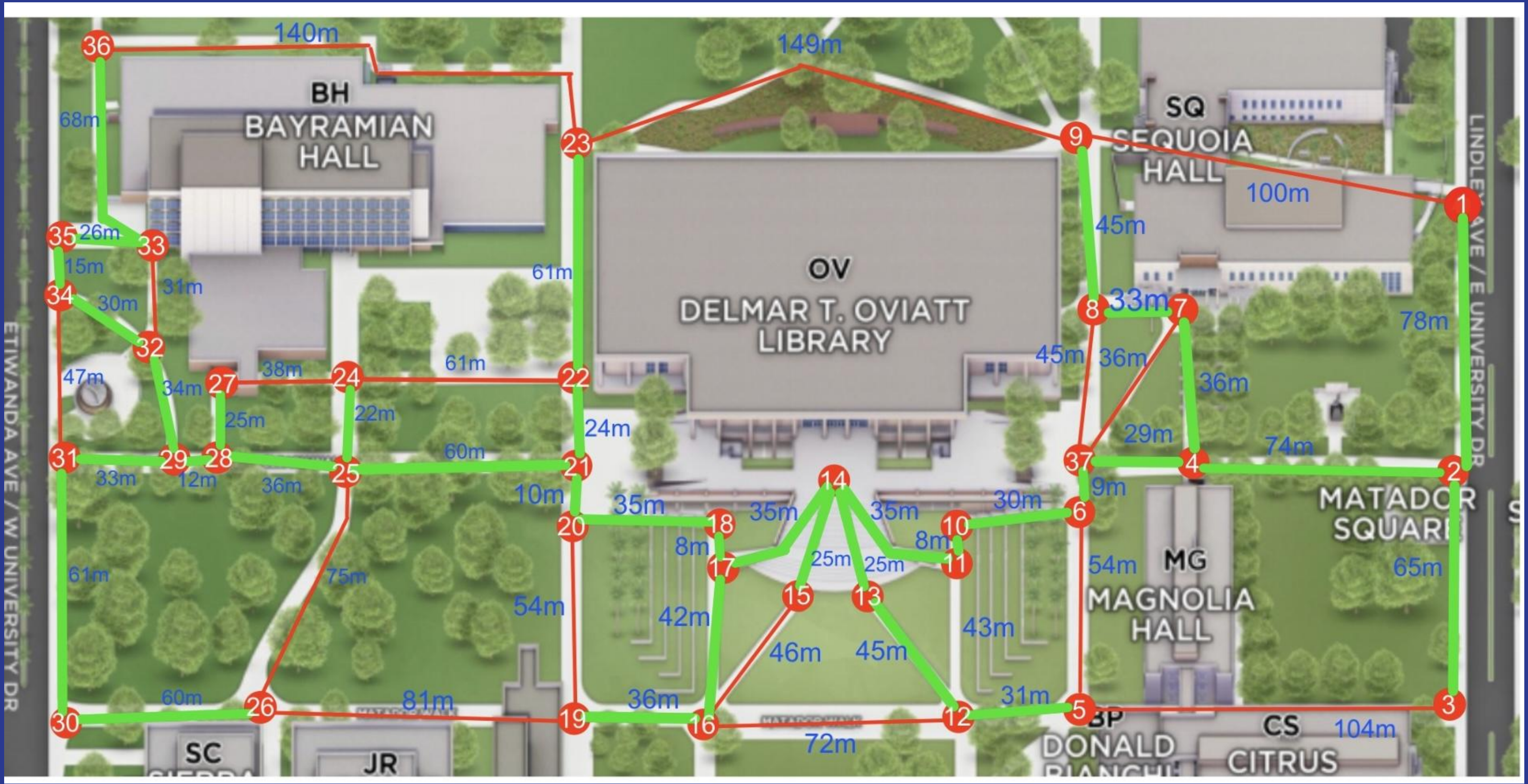


Figure 14 Showing minimum spanning tree of the data set collected

Vertices = 37

Edges 37 – 1 = 36

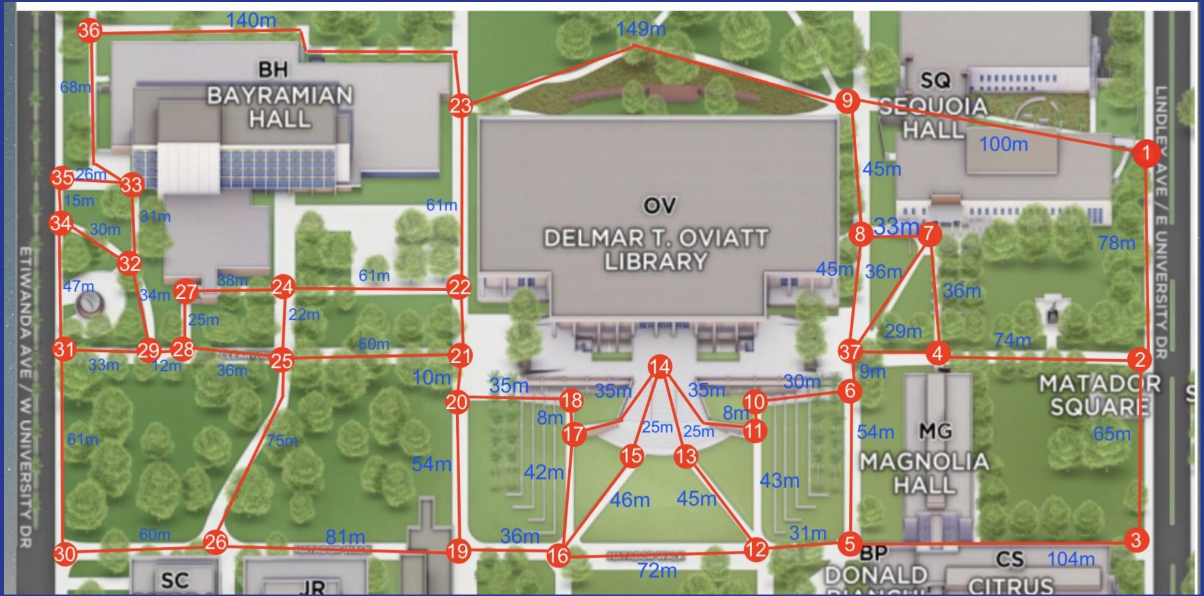
Final edges = 36

Minimum weight
1301 m

Figure 15 Showing minimum spanning tree of the data set collected after running Kruskal’s algorithm

Comparison between Kruskal’s algorithm & Prim’s algorithm

Data set for Prim’s algorithm same as kruskal



INITIAL GRAPH

Edge	Weight
1 - 2	78
2 - 3	65
2 - 4	74
12 - 5	31
37 - 6	9
4 - 7	36
7 - 8	33
8 - 9	45
6 - 10	30
10 - 11	8
13 - 12	45
14 - 13	25
11 - 14	35
14 - 15	25
17 - 16	42
14 - 17	35
17 - 18	8
16 - 19	36
18 - 20	35
20 - 21	10
21 - 22	24
22 - 23	61
25 - 24	22
21 - 25	60

Edge	Weight
30 - 26	60
28 - 27	25
25 - 28	36
28 - 29	12
31 - 30	61
29 - 31	33
29 - 32	34
35 - 33	26
32 - 34	30
34 - 35	15
33 - 36	68
4 - 37	29

Figure 16 Data set collected using Google maps

RESULTS

New Graph after applying Prim’s

Vertices = 37 **Edges 37 – 1 = 36** **Final edges = 36** **Minimum weight 1301 m**

Figure 17 Showing minimum spanning tree of the data set collected after running Prim’s algorithm

Comparisons and conclusions

Kruskal's Algorithm Results:	Prim's Algorithm Results:
Vertices = 37 Edges 37 – 1 = 36	Vertices = 37 Edges 37 – 1 = 36
Final edges = 36	Final edges = 36
Minimum weight = 1301 m	Minimum weight = 1301 m

Chart 1 compares both algorithms running in the same PC

Kruskal’s Algorithm Results:	Prim’s Algorithm Results:
• Dell PC 8.00 GB RAM	• Dell PC 8.00 GB RAM
• Intel(R) Core(TM) 2.30GHz	• Intel(R) Core(TM) 2.30GHz
• Run time: 0.025 sec	• Run time: 0.053 sec
• Big O = $O(E \log V)$	• Big O = $O(V^2)$

Chart 2 compares advantages & disadvantages for both algorithms

Kruskal's Algorithm	Prim's Algorithm
Greedy Algorithm	Greedy Algorithm
Often utilizes priority queue to select edges	Often uses a disjoint-set data structure to efficiently merge trees
Tens to be more efficient on dense graphs	Tens to be more efficient on sparse graphs

Contributions.

- Whit this project many tasks can be done at any time of the day even during night time when the campus is free of people and the robots can move freely.
- The school will save time and money to perform different activities.
- Students from other science majors can make improvements to this project.
- Will stimulate and promote research if it is implemented and improved.

Limitation and future work

1. Graph the entire CSUN campus in the future
2. Account for human traffic to estimate more accurate travel time
3. Expand dataset with unmarked paths to find shorter routes
4. Changes in terrain and obstacles might cause a static minimum spanning tree to become outdated.
5. Consider the line-following robot’s specific needs: avoiding steep inclines, etc.

References

[1] Restaurant Serving Robot with Double Line Sensors Following Approach. [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/8816404. [Accessed 19 Oct 2023]

[2] Line Follower Robot: Design and Hardware Application. [Online]. Available: https://ieeexplore.ieee.org/document/9197968. [Accessed 19 Oct 2023]

[3] Design and Development of Automated Guided Vehicle with Line Follower Concept. [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/10179618. [Accessed 21 Oct 2023]

[4] Exploring the parallel implementations of the three classical MST algorithms. [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/7975216. [Accessed 21 Oct 2023]

[5] Clifford A., Shaffer. "Kruskal's Algorithm". A Practical Introduction to Data Structures and Algorithm Analysis. 2nd ed. 2020 New Jersey, 2018, pp. 241, 386 - 387, 390.

[6] Weiss, Mark. "Kruskal's Algorithm". Data Structures and Algorithm Analysis. 2nd ed. 1994 California, 1994, pp. 312 - 314.

[7] Clifford A., Shaffer. "Kruskal's Algorithm". A Practical Introduction to Data Structures and Algorithm Analysis. JAVA ed. 1998 New Jersey, pp. 217 - 218, 222, 249.

References

[8] California State University Northridge CSUN,” 3dmap.csun.edu. [Online]. Available: https://3dmap.csun.edu/?id=1100. Accessed 15 Oct 2023

[9] Kruskal's Minimum Spanning Tree (MST) Algorithm [Online]. Available: https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/. Accessed 15 Oct 2023

[10] An Advanced Graph Embedding Framework with Node Embedding to Design Water Pipeline Networks [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/10330890. Accessed 15 Oct 2023

[11] Deterministic Energy Conserving Algorithms for Wireless Sensor Networks [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/1673166. Accessed 15 Oct 2023

[12] An algorithm for Hierarchical Chinese postman problem using minimum spanning tree approach based on Kruskal's algorithm. [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/7154702. Accessed 15 Oct 2023

[13] KMST: Kruskal’s Minimum Spanning Tree Algorithm. Google Scholar. https://link.springer.com/chapter/10.1007/978-981-10-8968-8_10. Accessed 15 Oct 2023

Q&A