TAYLOR'S UNIVERSITY
Wisdom · Integrity · Excellence

SCHOOL OF COMPUTER SCIENCE

BACHELOR OF COMPUTER SCIENCE (HONORS)

JANUARY 2023 – SEMESTER 1 (YEAR 1)

MODULE CODE:	**ITS 66204**
MODULE NAME:	**DISCRETE STRUCTURES**
MODULE LEADER:	**MR. SAJEEN TANDUKAR**
DEADLINE:	**29TH MARCH 2023**

GROUP PROJECT

Student Declaration: We Declare That –

- **I confirm my awareness about the university's regulations, governing cheating in tests and assignments, and form the guidance issued by the school of computing and it concerning plagiarism and proper academic practices, and the assessed work now submitted is in accordance with this regulation and guidance.**
- **I understand that, unless already agreed with the school of computing and it, that the assessed work has not been previously submitted, either in whole or in part, in this or any other institution.**
- **I recognize that should evidence emerge that my work fails to comply with either of the above declarations, then i may be liable to proceeding under regulation.**

GRAPH AND TREE

STUDENT FULL NAME	UNIVERSITY ID	SIGNATURES	SCORES
Sujal Ratna Tuladhar	036 2948		_____________ / 100

Table of Content

Table of Figures

1. Content

1.1. Introduction

1.1.1. Graphs and Trees

Public transportation systems is well known by all of the citizens in a country, but the study of their design from a topological/geometric perspective remains relatively limited. The findings of Euler in the mid-1700s discovering of networks routes. To the evolution of the topic from late 1990s to current ere. The goal of the report is to review the concepts of graph and trees, and literature applied in theory to make network. These are used in tools as GIS and GPS.

1.1.2. Keywords

Graph: A graph G is a set of nodes (vertices) V, connected by links (edges) E. Thus G = (V, E).

Vertex (Node): A node V is a terminal or an intersection point in a graph. It is the considered as a location such as a city, intersections or a terminal (stations, airports, and so on). They are typically the end points of edges and occur mid-link.

Edge (Link): An edge E is a link between nodes. The link (i, j) is for start i and of end j. A link/edge is a line segment representing the flows of movements between nodes. It has a direction that is commonly represented by an arrow. When not used, it is assumed as bi-directional.

Route/Path/Chain: series of connected links, travelling sequentially in a direction uninterrupted, with common connections.

Networks: A system of nodes and links which may consists of several model.

Note: In analysis of transportation network graph, they are always finite due to constraining boundaries.

Sub-Graph. A sub-graph is a subset of a graph *G*, *G'* = (*v'*, *e'*) can be a distinct sub-graph of *G*.

Directed Graph: flow of direction is shown definitive.

Undirected Graph: no flow of direction is indicated.

Cycle, Buckles, Loop, Self-Edge: flows from initial and terminal node into or within itself, link that corresponds to itself

Planar: when all edges meet at a node i.e.; vertices are linked. the intersections of two edges are a vertex and its topology is two-dimensional.

Non-Planar: when edges may cross paths; when links intersect but there are no vertices at the intersection of at least two edges. Here, it works with a third dimension since there is the possibility of a movement over-head another as for air and marine transport, and overpass/ bridge roads.

Simple graph: A graph that has only one type of link between its nodes i.e., it here has no loop or parallel edges.

Multigraph: A graph that constitute of several types of links between its nodes. Some has one other may have multiple.

Element: a graph cell (dyad)

Topologic Distance/Length/Connection: a measurement of distance between nodes which are unit less. They are steps where increase in steps means longer distance. These measurements aren't used in real life. Numbers of links in terms of traffic or others.

Circuit: a path of corresponding nodes, where links are travel in same directions, it covers distribution territory i.e., the delivery route.

Diameter: the number of links needed to connect two furthest nodes in a network. They must be the shortest possible route. Where loops, detours and backtracking aren't included. Also dubbed as 'longest, shortest path'. In other words, it is the maximum numbers of steps to connect two points in a graph.

Root: A root is a node (R) which is the starting point of a distribution system path where every other node is coming from.

Trees: A connected acyclic simple graph without a circuit cycle. If a node is to be added the cycle is complete but if it is removed the link is destroyed.

Forest: a graph that contains no cycles, but doesn't need to be connected like a tree on n vertices, aka group of 'smaller' trees.

1.2. Background of Study

A graph is a symbolic representation of a network and its connectivity, The origin of graph theory can be traced to 1735 in Königsberg, Prussia (today's Russia). where Leonhard Euler (1741), came across a problem known as the "Seven Bridges of Konigsberg". Here someone could only cross all the bridges once, in a continuous sequence, Euler proved the problem that seemed to have no solution by representing it as a set of nodes and links. This led to the foundation of graph theory and its gradual improvements. It has evolved in the last decades and century by growing influences from studies of complex networks in all sort of topic.

In transportation geography, most networks have structural groundwork, such roads, transits, and rail networks, which are more known by their links than by their nodes. Which is not true case for all transportation networks. For particular sample, aquatic/naval and air/sky networks tend to be more defined by their nodes than by their links since the paths aren't clearly defined. A telecommunication route system can also be represented as a media network, would be difficult to represent. The most complex graphs to be considered are of networks or the Internet, may have structure that can be difficult to symbolize. Other way around, phones can be represented as nodes, while the links could be calls. Similarly, servers can also be expressed as nodes, while the physical infrastructure in between like fiber optic cables, could be links. So, all networks can be embodied by graph theory in hook or crook.

More than a century later, in the late 1800s, the British mathematician Cayley (1889) used a graph theory approach to study trees, leading to Cayley-trees. Trees are graphs without cycles or loops. Cayley's Formula tells us how many different trees we can construct on n vertices. We can do this with n vertices and then placing edges to make a tree. Another way is with a complete graph on n vertices, then removing edges in order to make a tree. Cayley's formula tells us how many different ways we can do this. These are called spanning trees on n vertices, and we will denote the set of these spanning trees by Tn.

1.3. Problem Statement

- Design a map using graph theory, analyze the connectivity of places in this vehicle network and demonstrate how theory can be used to analyze transportation networks.
- Understand the properties like connectivity and efficiency and analyze the impact of new roads.
- Problems involving effective methods for locating the best pathways in graphs (based on various criteria)
- The decision dilemma is not presented as a single decision (since the decision we make today Affects the decision we make tomorrow) or even as a series of decisions (because under uncertainty, decisions taken in the future will be influenced by what we have learned in the meanwhile). The issue is presented as a decision tree. (Magee)

1.4. Objective of Study

- Finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its individual edges is reduced is known as the shortest path problem and its efficiency. ("Shortest Path Problem")
- Finding a spanning tree with the fewest types of labels is the goal of the minimum labeling spanning tree issue, which arises when each edge of a graph is assigned a label rather than a weight. ("Minimum Spanning Tree")
- To apply graph theory in real-world problems, such as network analysis, scheduling problems, and optimization.

- To study various graph algorithms, such as Dijkstra's shortest path algorithm, Kruskal's minimum spanning tree algorithm, and Prim’s algorithm

2. Methodology

Introducing keywords, the origins and the history of the topic, we review the current developed road map to study transport networks of our city and modeled it as a graph. Afterwards, we examine indicators and characteristics of maps developed to study the network systems. in particular we review the areas we are very familiar with. We used the concepts of nodes to pinpoint the locations in the city, road/routes to draw a line joining the nodes which is the edge, etc., to develop a network over an already existing. Finally, we identify challenges to be addressed. The properties of this graph can be analyzed to study various aspects of the transportation network, such as the connectivity of cities, the efficiency of routes, and the impact of new roads or routes on the existing infrastructures.

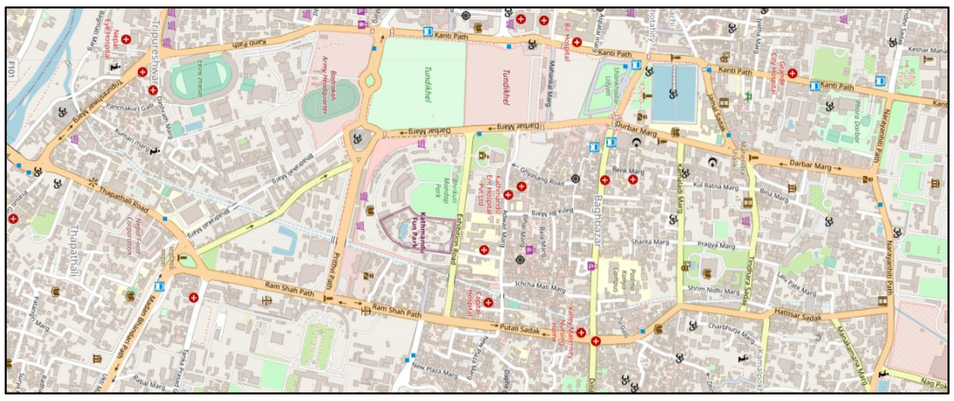


Figure 1. Road System of Inner/Central Kathmandu without Nodes and Paths

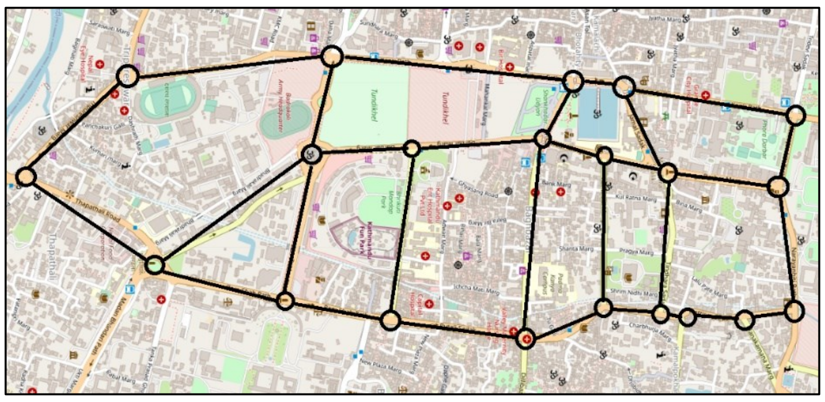


Figure 2. Road System of Inner/Central Kathmandu with Nodes and Paths

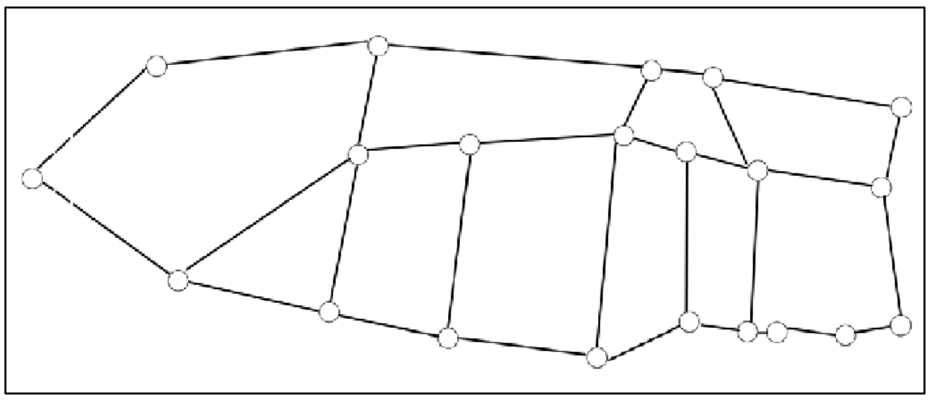


Figure 3. Outline of Nodes and Paths without the background

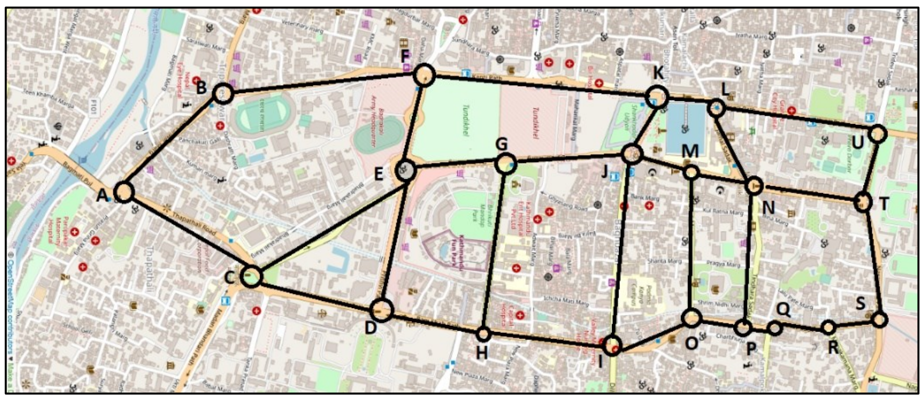


Figure 4. Map with Edges and Vertices

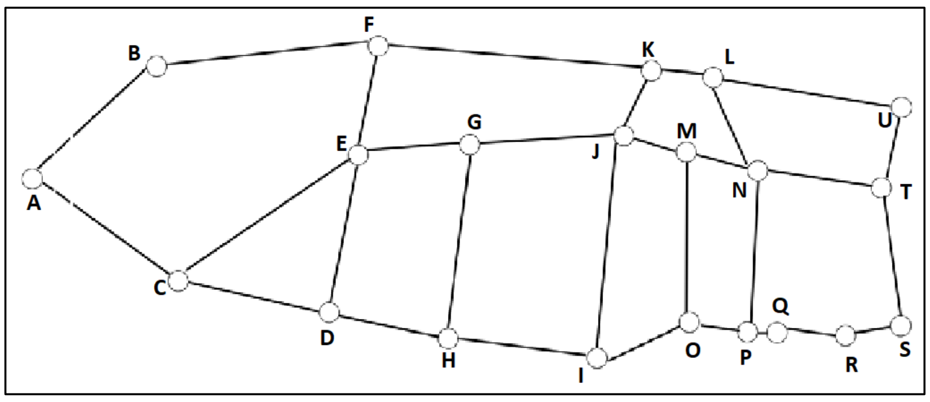


Figure 5. Map with labeled nodes

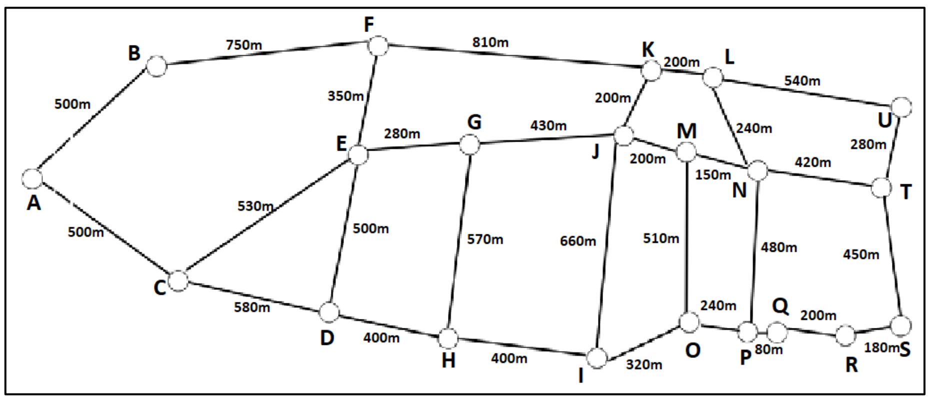


Figure 5. Map with Distance (in m)

Figure 6. Map with only nodes

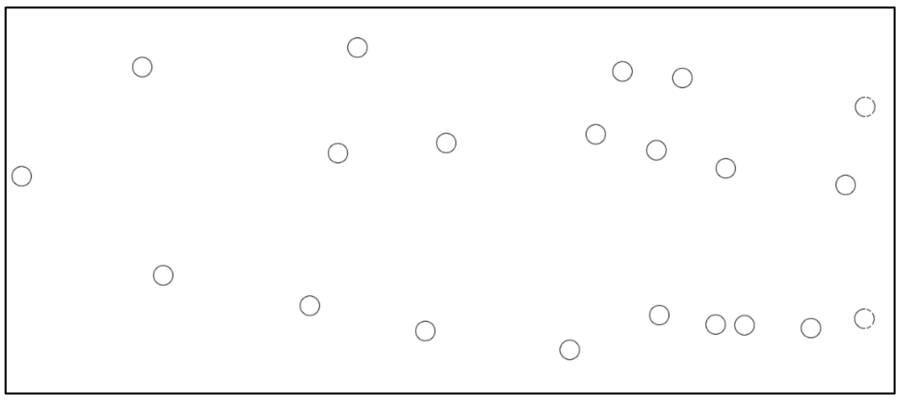


Figure 7. Nodes map

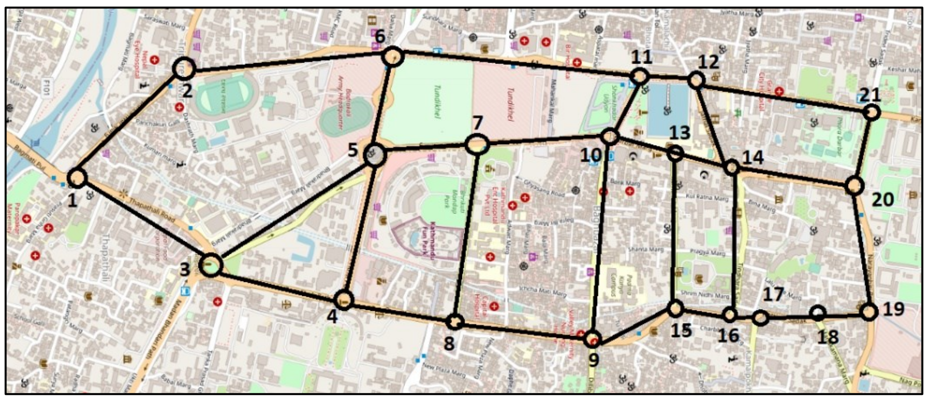
3. Result

G = (V, E); V = set of vertices, E = set of edges.

V = {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U}

E = {(A,B), (A,C), (B,F), (C,D), (C,E), (D,E), (D,H), (E,F), (E,G), (F,K), (G,H), (G,J), (H,I), (I,J), (I,O), (J,K), (J,M), (K,L), (L,N), (L,U), (M,N), (M,O), (N,P), (N,T), (O,P), (P,Q), (Q,R), (R,S), (S,T), (T,U)}

Order of Graph: |V| = 21



Size of Graph: |E| = 30

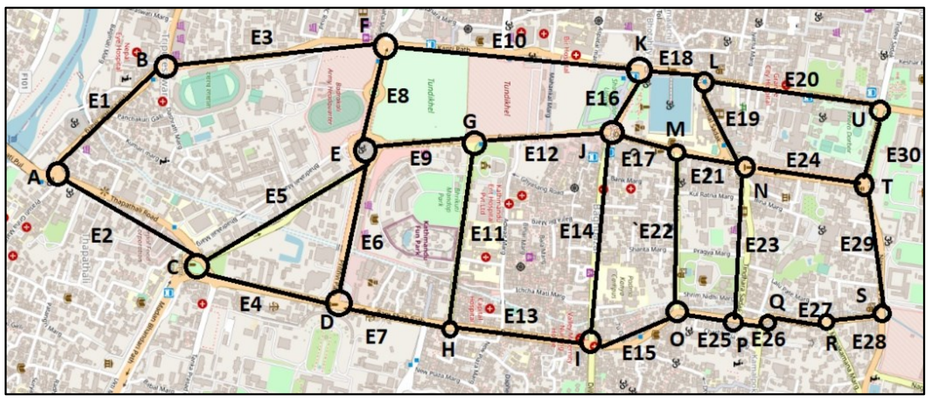


Figure 8. Map with labeled Nodes and Edges

3.1. Adjacency:

1	2	3			Null
2	1	6			Null
3	1	4	5		Null
4	3	5	8		Null
5	3	4	6	7	Null
6	2	5	11		Null
7	5	8	10		Null
8	4	7	9		Null
9	8	10	15		Null
10	7	9	11	13	Null
11	6	10	12	14	Null
12	11	14	21		Null
13	10	14	15		Null
14	12	13	16	20	Null
15	9	13	16		Null
16	14	15	17		Null
17	16	18			Null
18	17	19			Null
19	18	20			Null
20	19	21			Null
21	12	20			Null

3.2. Matrix:

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
01		1	1																		
02	1					1															
03	1			1	1																
04			1		1			1													
05			1	1		1	1														
06		1			1						1										
07					1			1		1											
08				1			1		1												
09								1		1					1						
10							1		1		1		1								
11						1				1		1	1								
12											1			1							1
13									1	1				1	1	1					
14												1				1				1	
15									1				1	1		1					
16														1	1		1				
17																1		1			
18																	1		1		
19																		1		1	
20														1					1		1
21												1								1	

3.3. Minimum Spanning Tree

3.3.1. Prim's Algorithm

- It is a greedy algorithm as it starts with empty spanning tree maintaining sets of vertices in the graph known as a cut, every step the edge with minimum weight is picked
- Starts from a single node and moves through adjacent nodes to all other nodes by connecting all nodes by edges
- Determine the arbitrary vertex as start point, Find the edge connecting any free vertex
- Find the minimum edges among the graph
- Add it to the minimum spanning tree if it doesn't form any cycle

500+500+530+350+280+430+500+400+400+200+200+150+510+240+80+200+180+240+420+280=6590

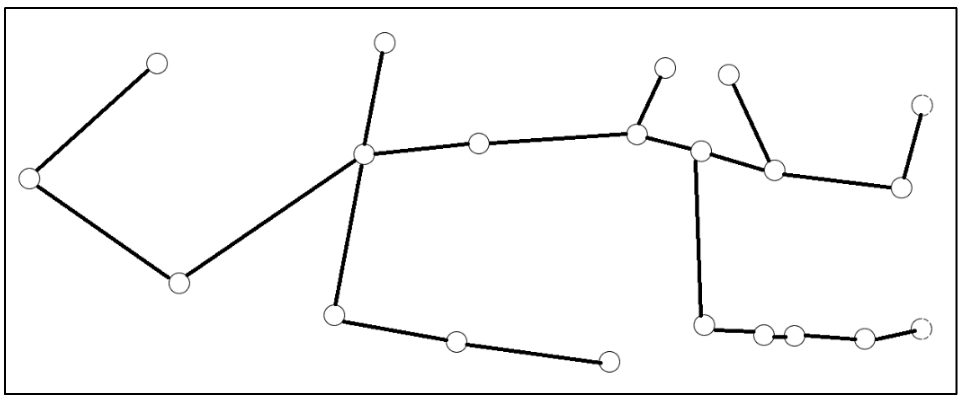


Figure 9. Prims Path outline

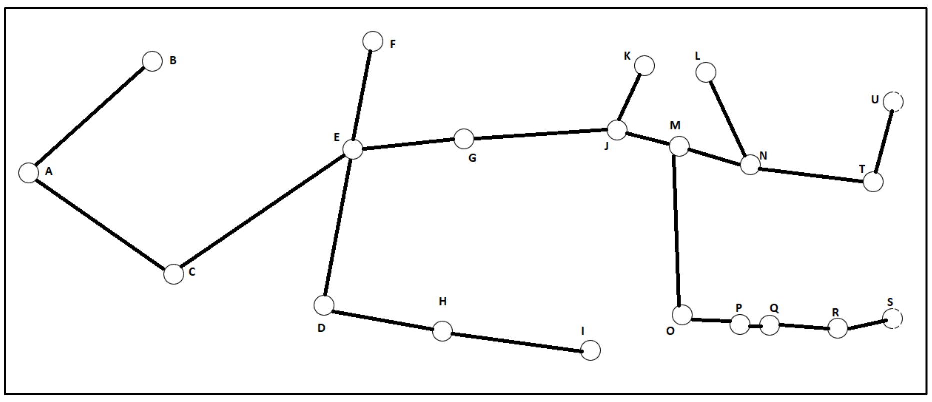


Figure 10. Named path outline

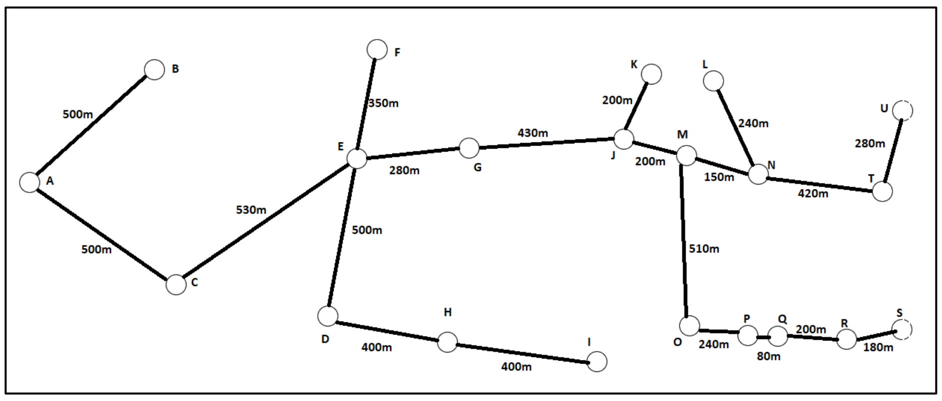


Figure 11. defined edges

3.3.2. Kruskal’s Algorithm

- All edges in the given graphs are sorted in increasing order, New edges and nodes are added if it doesn’t form a circle
- Picking minimum weighted edge first and maximum edge weight at last which is a logically optimal choice in each step

Edge No.	Vertex Pair	Edge Weight
E26	(P, Q)	80
E21	(M, N)	150
E28	(R, S)	180
E16	(J, K)	200
E17	(J, M)	200
E18	(K, L)	200
E27	(Q, R)	200
E19	(L, N)	240
E25	(O, P)	240
E9	(E, G)	280
E30	(T, U)	280
E15	(I, O)	320
E8	(E, F)	350
E7	(D, H)	400
E24	(N, T)	420
E12	(G, J)	430
E29	(S, T)	450
E13	(H, I)	470
E23	(N, P)	480
E1	(A, B)	500
E2	(A, C)	500
E6	(D, E)	500
E22	(M, O)	510
E5	(C, E)	530
E20	(L, U)	540
E11	(G, H)	570
E4	(C, D)	580
E14	(I, J)	660
E3	(B, F)	750
E10	(F, K)	810

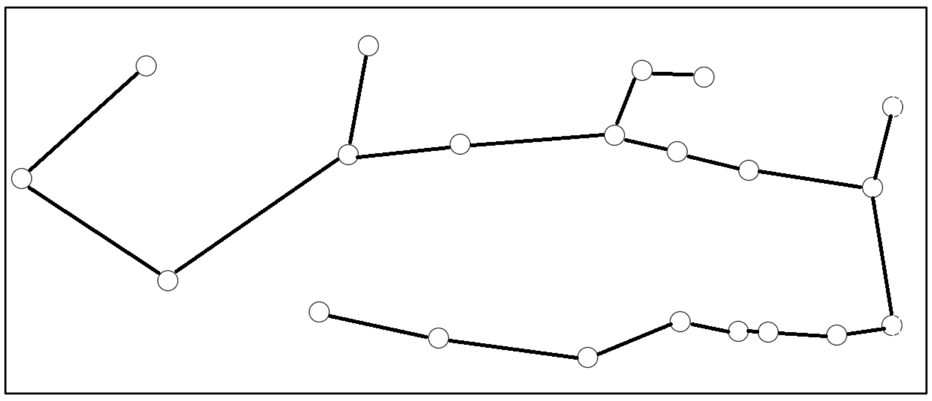


Figure 12.Kruskals Path Outline

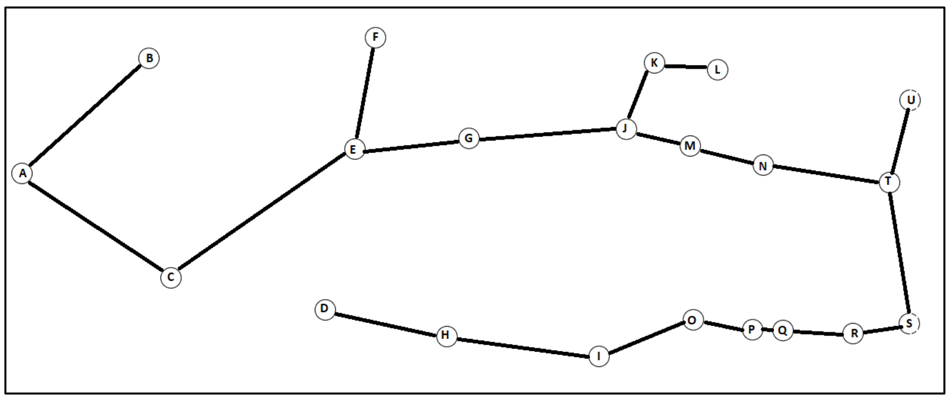


Figure 13. Named Path outline

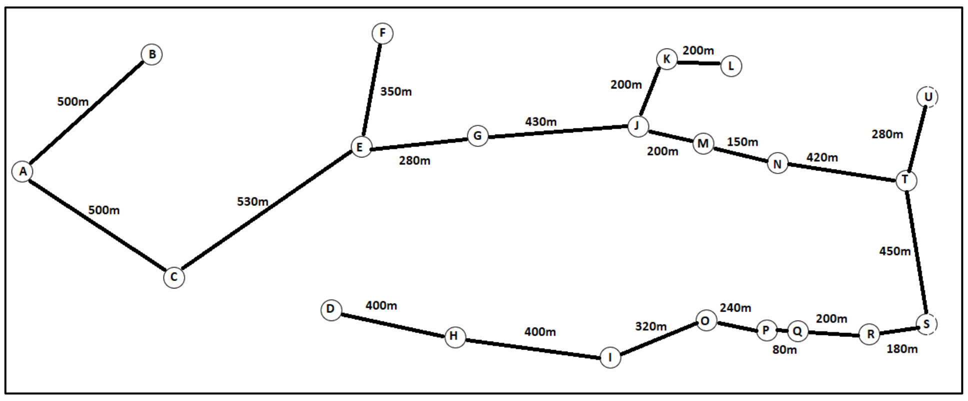


Figure 14. Weighted Edges

3.4. Shortest Paths

3.4.1. Dijkstra algorithm

Differs from Prim’s and Kruskal’s to generate shortest path tree with a root which include the vertices including the shortest path

- Shortest path from one node to every other node
- Keep list of unvisited nodes and keep track of distances
- Assign value 0 to initialization point then pick a point with minimum distance values,
- for every adjacent vertex, calculate the sum of distance and
- if weight of vertex is less than the sum vertices, update the distance as the value of weight

- but if the sum of vertices is less than the weight updates the distance as the value of sum

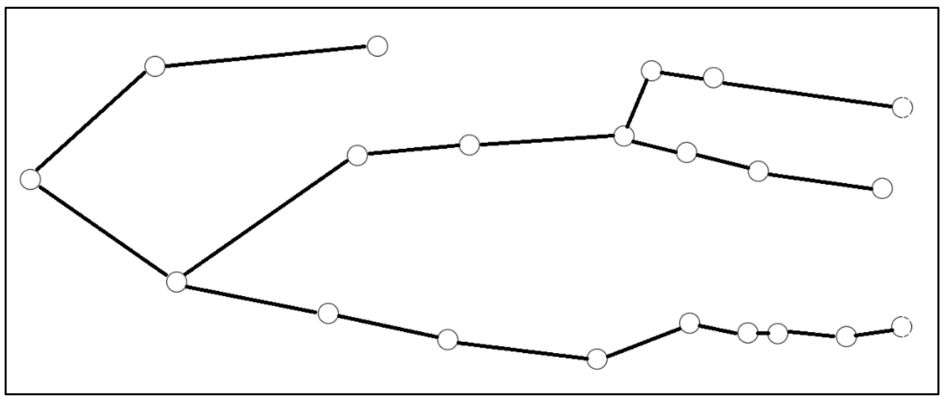


Figure 15. Dijkstra Path Outline

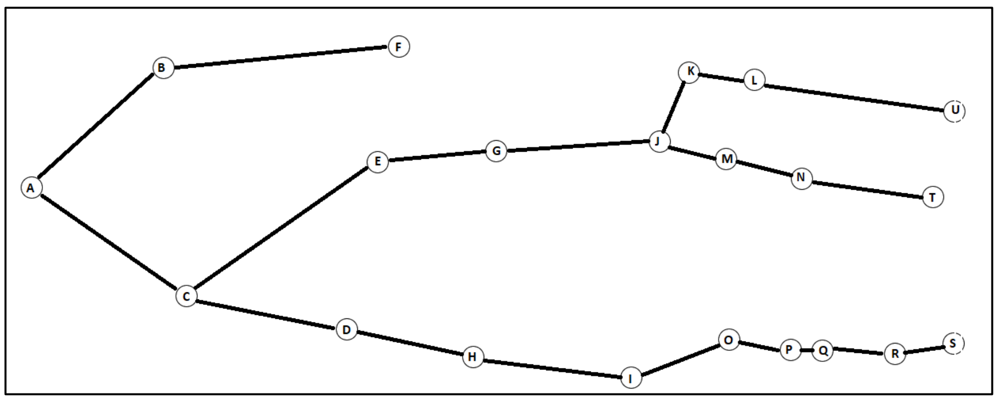


Figure 16. Named Path Outline

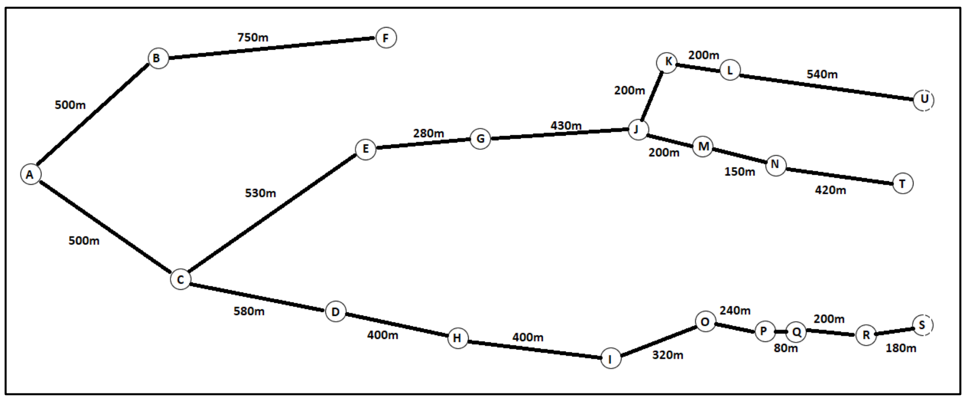


Figure 17. Weighted Paths

4. Limitation

The limitations of graph algorithms

- graph theory is a spatial analysis with variable ecological relevance's; arcs, path lengths, structures, nodes, orders, shapes, minimum spanning trees
- significance loss of information due to multi-layered thresholds
- computational limitations and human errors
- can only represent relationships of objects not physical properties or attributes
- difficult to interpretate large and complex, cannot extract meaning full data which require highly advanced analytical techniques and expertise
- requires more effort and power to process and analyze large and complex data
- multiple varieties of graph technique with its own strengths and weaknesses

5. Recommendation

we recommend using graph algorithms

- to represent complex data; effective tool to represent complex data
- to perform and process complex operations efficiently using specifically designed structures
- to find the shortest path between two points; problem faced in transportations and logistics
- to visualize complex data in concise way and make meaning full presentations and report analytics

6. Conclusion

The goal of this report was to review and evaluate the current transportation network design from a topological perspective. The graph theory is used to address problems of network designing. The origin of graph theory from the works of Euler is old, dating back to the 18th century. However, much later in the mid to late 1900s, Applications of transit systems was developed with network indicators.

The study of transit networks using a graph theory approach emerged in the 1980s. different indicators have been developed over the course of centuries. Specific characteristics were also developed to better understand the topologies of networks. The field to analyze transit networks is looming strongly and seems promising under network science.

This research report can be particularly useful to any who are looking to familiarize themselves with the network graph. Getting an understanding of such properties of transportation can be of great help for future purposes, and this subject matter becoming even more relevant in the world are likely to grow due to exponential growth in population.

Nevertheless, much work lies ahead and we have identified three challenges that will need to be solved. First, one standard methodology to study transit systems as graphs/networks should be developed. Second, the fact transit systems have different modes should be accounted for to solve issues of modal integration. Third, a comprehensive list of network design indicators is needed to assess and guide network design, which could be particularly helpful to transit planners and engineers in the world.

7. References

Arlinghaus, S.L., W.C. Arlinghaus, and F. Harary (2001) Graph Theory and Geography: An Interactive View. New York: John Wiley & Sons.

Casarotto, C. (2006). Graph Theory and Cayley's Formula. https://math.uchicago.edu/~may/VIGRE/VIGRE2006/PAPERS/Casarotto.pdf

Garrison, W. and D. Marble (1974) "Graph theoretic concepts" in Transportation Geography: Comments and Readings, New York: McGraw Hill, pp. 58-80.

Morlok, E. K. (1970) An Analysis of Transport Technology and Network Structure (Evanston, IL: Transportation Center Northwestern University).

Perfect, H., Biggs, N. L., Lloyd, E. K., & Wilson, R. J. (1977). Graph Theory 1736-1936. The Mathematical Gazette, 61(417), 233. https://doi.org/10.2307/3617244

Rodrigue, J.-P. (2020). The Geography of Transport Systems, Fifth Edition, New York: Routledge. Journal of Transport Geography, 17(3), 246–247. https://doi.org/10.1016/j.jtrangeo.2009.01.007

Rosen, R. (1962). The theory of graphs and its applications. The Bulletin of Mathematical Biophysics, 24(4), 441–443. https://doi.org/10.1007/bf02478000

Voss, W. (1979). Andrásfai, B., Introductory Graph Theory. Budapest. Akadémiai Kiadó. 1977. 268 S. ZAMM - Zeitschrift Für Angewandte Mathematik Und Mechanik, 59(8), 401–401. https://doi.org/10.1002/zamm.19790590824

Weisstein, E. W. (n.d.). Wolfram MathWorld: The Web's Most Extensive Mathematics Resource. Mathworld.wolfram.com. https://mathworld.wolfram.com/