Addis Ababa Institute of Technology
School of Information Technology and Engineering
Department of IT/SE Eng.

Artificial Intelligence Assignment

Name: Henok Mekuanint
Section: 1
ID:NO: UGR/2272/12

Instructor:Amanuel Negash

1.NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the communication and interaction of human language and a Computer System and its ability to understand human voice and text speech. Natural Language Processing gives a special emphasis on the recognition of human language by different machine. Natural Language Processing also includes the respondence for the text of voice that the machine recognized.

Natural Language Processing doesn't follow a mere sequence of symbols but it rather understands the structure of language for example the hierarchical structure of a language is that words make phrases, phrases make a sentence and the sentences shows ideas. Natural Language Processing is used in different sector some of them are

- Text mining
- Machine translation
- Automated question answering

What are the stage in natural language processing ?

Natural Language Processing has 5 major stages.

1. **Morphological Analysis/Lexical Analysis:**

   It is the study of structure of words. It identifies how a word is produced through the use of morphemes. The fundamental unit of Natural Language Processing that cannot be sub divide further.

2. **Syntax Analysis**

   The syntax Analysis ensures that a given piece of text is correct. The Purpose of this stage is to draw the exact meaning from the text the user provided.

3. **Semantic Analysis**

   The semantic analysis starts with lexical semantics which individual words meanings. Semantic analysis then examines relationships between individual words and analyzes the meaning.

   Semantic Analysis of Natural language processing captures the meaning of the given text while taking into account context,logical structuring of sentences and grammar roles.

4. **Disclosure**

   Disclosure is the effect of a previous sentence on the sentence for example we can point to an object that is previously mentioned on our sentence using pro nouns so the NPL must recognize this pro noun and to whom it is referring to.

5. **pragmatics**

This is the last step in Natural Language processing. This step interprets the given text using information using the information from the previous steps. Given a sentence ,” shut the door” is an order.

1.NATURAL LANGUAGE PROCESSING

Agent Type	Performance measure	Environment	Actuator	Sensor
Natural Language processing	To understand human voice and text, to provide necessary information based on the input	-Security guard -Voice recognizer application -Translator in international meeting	-Scan report -Screen Display From	-Camera -Vibration Sensor or audio sensor are used -Infrared -Sensors

Performance measure of NLP -

The performance measure of NLP is measured by the ability to understand human voice and text and provide necessary information as soon as possible. Shazam is an app that recognizes human voice and it has the ability to search on the internet the song that it recognized it uses a Natural Language processing in order to recognize the voice and it will convert it to text after that it will search the song that perfectly matches with the lyrics the NLP recognized.

The second one is a Grammar checker, a grammar checker. In order to check the grammar the NLP need to have the ability to recognize the semantic analysis of the sentence or the phrase that is written.The disclosure and syntax analysis must also be considered so that it will provide the necessary grammatical suggestions. Over all the performance measure of a Natural Language Processing in determined by the ability to understand and recognize the text or speech.

Environment -

NLP can be used in many environments from hospitals and security places that need voice recognition to apps. Now days due to globalization interaction and communication of people with different language has increased, NLP can be used as a translator between different languages.

Actuator -

Actuators are the component of machines that does physical tasks like moving . The NLP can make an output through different mechanisms for example it can make an output using a screen record to show suggestion of an incorrect grammatical sentence.

Sensor -

NLP has different sensors that take an input and process it and make the output through actuators. NLP can use camera vibration sensor or audio sensor or scanner to take an input from the environment and perform its ask.

2. Autonomous cars

Self driven car or autonomous car is a vehicle of sensing its environment and operating without human involvement. A human passenger is not required to take control of the vehicle at any time nor is a human passenger is required to be present in the vehicle.

There are six level of autonomous vehicles as the levels increases the extent of the driver-less car's independence regarding operation control increases.

Autonomous vehicles has several advantages some of the advantages are

- Less traffic jams
- stress free parking
- Time- saving vehicle

Agent Type	Performance measure	Environment	Actuator	Sensor
Automated car Drive	The comfortable trip, Safety, Maximum Distance	Roads, Traffic, Vehicles	Steering wheel, Accelerator, Brake,Mirror	Camera -GPS -Odometer

Performance measure of autonomous cars-

The performance measure for the automomus car will be the ability the ability to drive the car without the intervention of humas by using difference sensors and acutators. The autonomous car is a vehicle capable of sensing its environment and operating withput human involvement. An automomous car can go anywhere a traditional car goes and d everything that an experinced human driver does.

Environment

The environment measure for the autonomous car are Roads, Traffic ,vehicles. The autonomous car interacts with different environment especially with roads and vehichles from taking a forward step the car needs to consider its enviromnet inorder not to have avoid any kind of accident. While the autonomous car parks it self in needs to consider the environment in detal so that other cars can be near.

Actuator-

Autonomous car has different Actutatos to perform different movements for example motors sterring wheel,Accelarator,Brake,and mirror are some of the acutators.

Sensor-

Cameras GPS and Odometers act as a sensor because the autonomous cars percieve information from different environments .

Breadth First Search

The breadth first search is an algorithm for traversing or searching tree or graph data structures. It explores all the nodes at the present depth before moving on to the nodes at the next depth level. The breadth First can be implemented using a queue and Breadth first search doesnt consider weight while calculating the shortest path. Breadth first search algorithm starts the operation from the first or starting node in a graph and it traverses it thoroughly. Once it successfully traverses the initial node then the next non-traversed vertex in the graph is visited and marked.

Breadth First search is a level order search which means it search the Targer as a level order. It moves on level further for each iteration or recursion.

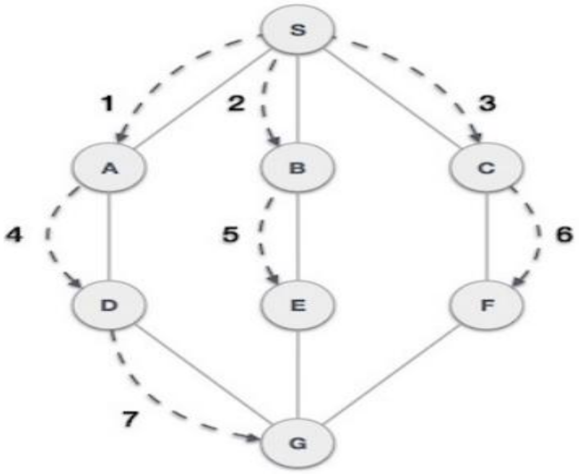


Figure 1.1

Observed Time and Space Complexity of BFS

- Since in the worst case breadth-first search has to consider all paths to all possible nodes the time complexity of breadth first search is O(|E|+|V|) where |v| and|E| is the cardinality of set of vertices and edges respectively.
- The Complexity is this since every vertext and every edge will be explored in the worst case.
- Since all of the nodes of a level have been generated the space complexity is proportional to the number of nodes at the deepest level.
- In The worst case sencario the graph has a depth of 1 and all vertices must be stored.

INPUT AMOUNT	Original readed graph 1 X	Twice of the graph 2 X	Tripeld of the graph 3 X
Average Time in microsecond	93.88250000000001	1669.8425	10204.257500000003
Average length	47.425	325.35	1207.3950000000004

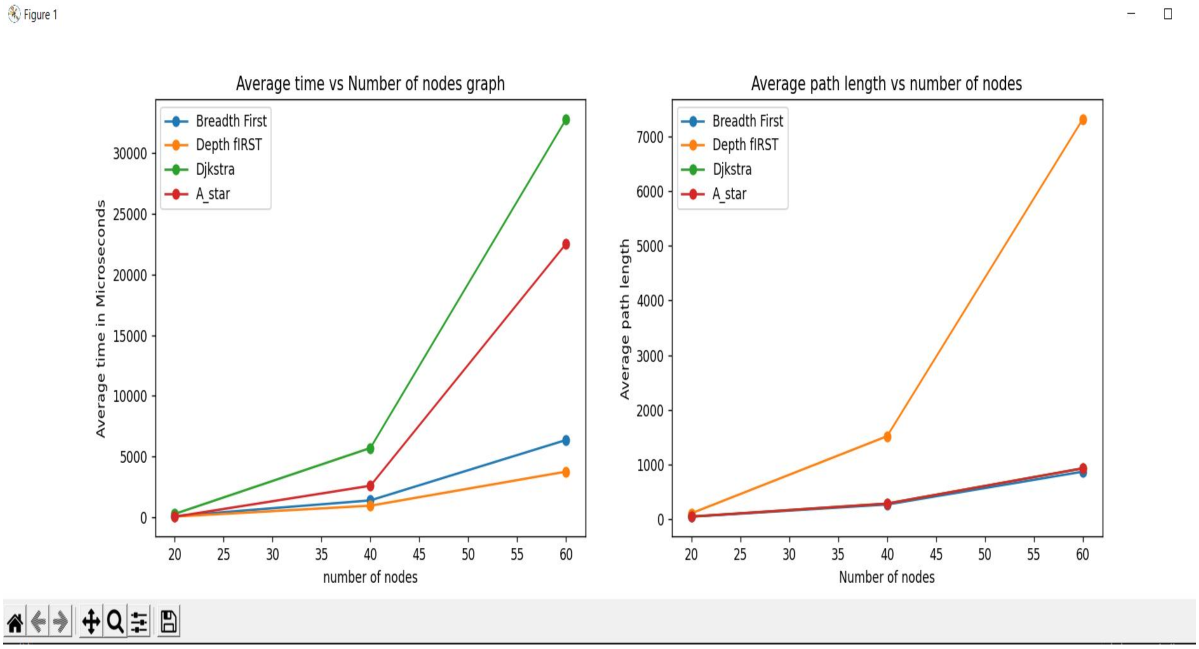
Average time analysis on BFS:

Since breadth first search is a level order search it checks all the possible path but it depends on the graph generation for exapmle if the graph is very dense rather than sparsely populated graph it will become slower than the other graphs but if it is more sparse and randomly connected it will become faster in pathfinding because it is a level order traversal. As the number of Random nodes generation increases. When the size of the graph became doubled or tripled the time to find the path will also increase very solowly compared to others.

Average length analysis of BFS:

According to my algorithm breadth first search has comparable path lengt with djkstra and a star search algorithms but it has a big difference with depth first search.
In the above Image the first graph shows that the average time in microsecond vs the number of nodes and the seccond graph shows the average path length vs thee number of nodes.
Since breadth first search doesn’t consider weight the path that is going to be found will not consider weight.

Obseration in graph:



Figuer 1.2 BFS analysis

Depth First search

Depth first search (DFS) algorithm traverses a graph in a depth motion and uses a stack to remeber to get the next vertex to start a search when a dead end occurs in any iteration. I have implemented the depth first search using Iteraive solution.

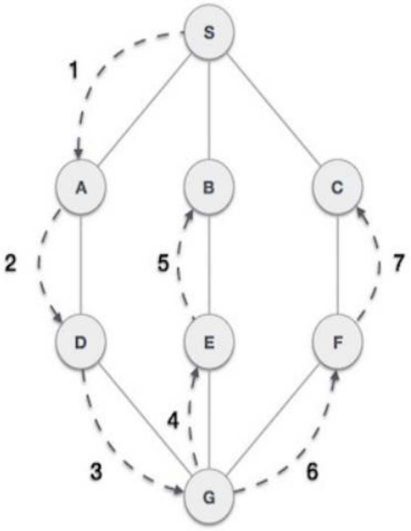


Figure 1.3

Oberved Space and Time Compexity of depth first search

- If the entire graph is traversed, the temporal complexity of DFS is O(V) where V is the number of vertices.
- The temopral time complexity is O(V)+O(E)=O(V+E)

Average time analysis of DFS:

Depth first search average time is found to be the smallest that means it is easily finding the destination shortly this is because the graph is sparcely rather that dense if it were dense DFS will take much longer than the sparcely because if the target node hasn’t been found it will trace back and search for it.

Average length analysis of DFS:

The average length of Dfs is largest than the other search algorithms because It visits unecessary nodes and when it finds that the path has no negighbour it backtracks . Increasing the size of the graph makes dfs average length to be larger.

INPUT	Original readed graph 1 X	Twice of the graph 2 X	Tripeld of the graph 3 X
Average Time in microsecond	55.707499999999996	838.2675000000002	6662.752499999998
Average length	109.725	1628.7749999999999	8943.3

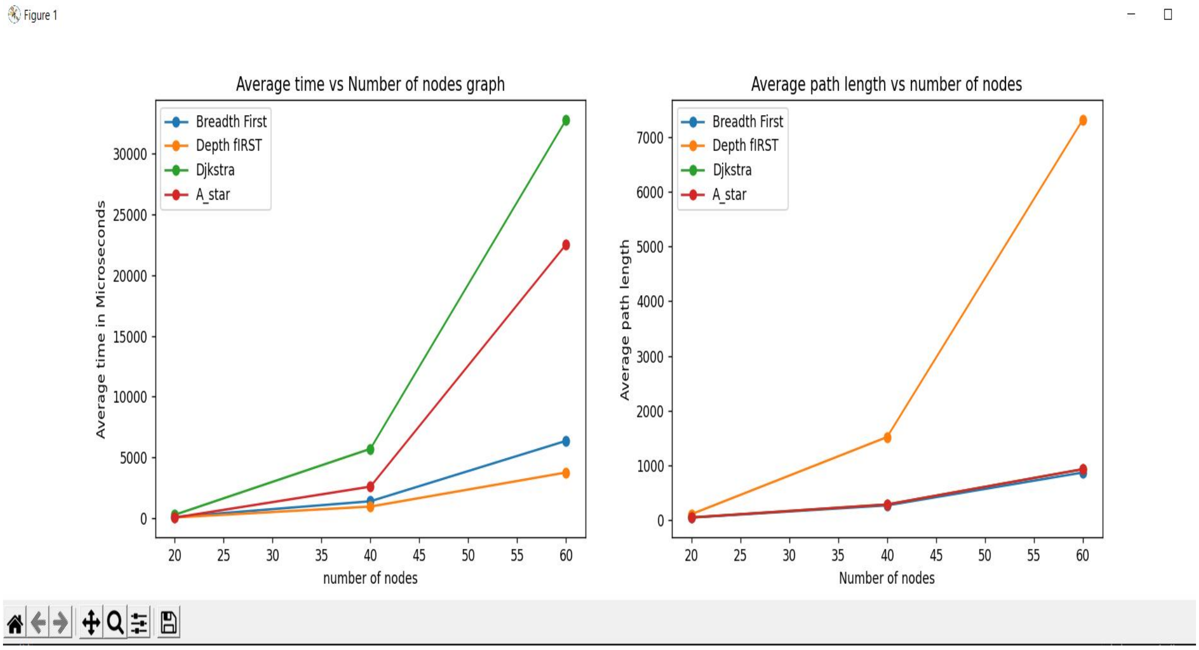


Figure 1.4

Djkstra Shortest path Algorithm

Djkstra algorithm is a single-source path algorithm. Single- source means that only one source is given and we have to find the shortest path from the source to all the nodes. Djkstra used this property in the opposite direction I.e we overestimmate the distance of each vertex from the starting vertex. Then we visit each node and its neighbours to find the shortest subpath to those neighbours.

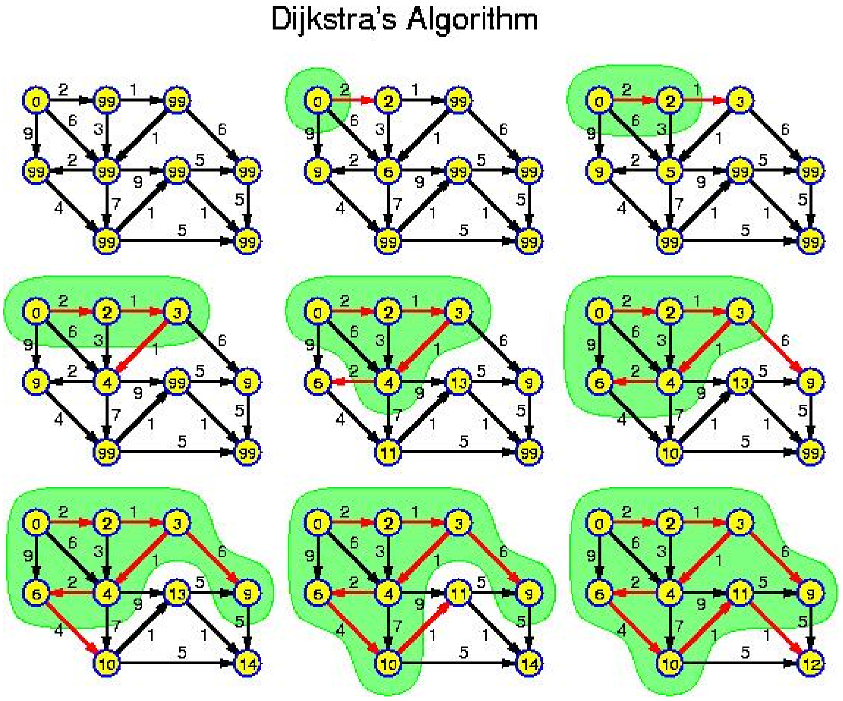


Figure 1.5

To implement djkstra each of the vertices will be originally marked as unvisited , it means that elements of the visited array are set to false. As we must find the most profitable paths, in each element of the distance vector path is recorded. As a starting vertex and its attributed to the zero path as there is no edge from s to s. I have used relaxation approach and thus finds the locally optiman choice at each step of the algorithm.

Mathematically relaxation can be represented as

$$\mathbf{P\,[\,V\,] = minimum\,(\,p\,[\,v\,]\,,\,p[\,n\,] + w\,)}$$

Where p [v] is the current path value for node v,p [n] is the path value upto the previously visited node n and w is the weight of the edge between the current node and previously visited node n ans w is the weight of the edge between the current node and previously visited node.

I have used two properties the Visited property and the path property. The visited property represents whether the vertex has been visited or not which is used not to revisit the vertexes again. The path property stores the value of the current minimum path to the vertex.

The observed time complexity the time complexity of djkstra algorithm is O (V^2) where V is the number of vertices in the graph.

- The time to check unvisited vertex with smallets path is O (V).
- The time required for relaxation will be O (1)

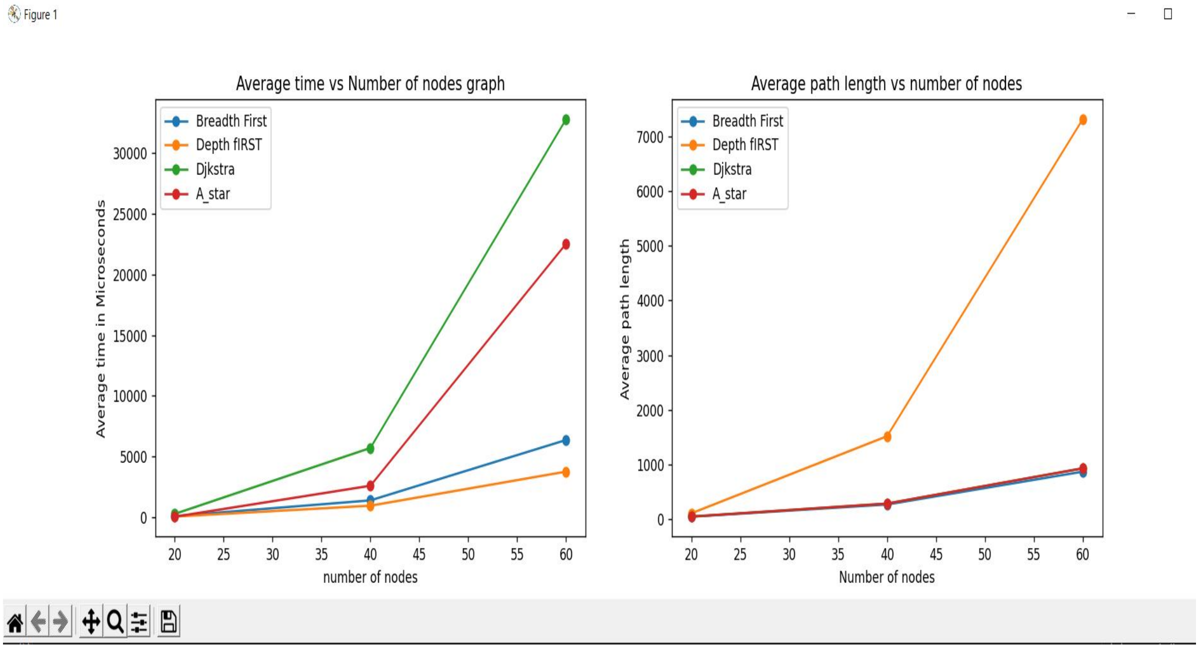
Average time analysis of Djkstra:

The BFS and DFS most of the time doesnt consider the weight of the graph betwen the two nodes but most often weight is necessary thing to consider. Considering the anakysis table Djkstra has been found to be the slowest than others while in path finding because the graph is sparse and big so djkstra always looks the shortest weight to travel so such kind of algorithm is a little bit slower when the input become larger .

Average length analysis of Djkstra:

Considering the average length djkstra became comparable with A star and BFS but it has a big difference with DFS since DFS will take so many steps to find the target node for such larger graph.

INPUT	Original readed graph 1 X	Twice of the graph 2 X	Tripeld of the graph 3 X
Average Time in microsecond	324.6	5770.725	40130.6
Average path length	49.74999999999999	341.99749999999995	1279.14



A * searching algorithm

A star algorithm works by making a lowest cost path free from the start node to the target node. A star algorithm is different because it uses a function that gives an estimate of the total cost of a path using that node. Therefore, A* is a heuristic function which differs from an algorithm in that a heuristic is more of an estimate and is not necessarily provalbe correct.

$$F(N) = G(N) + H(N)$$

Where

- F(n) = total estimated cost of path through node n
- G(n)= cost so far to reach node n
- H(n)=estimated cost from n to goal. This is the heuristic part of the cost function so it is like a guess.

The main observed difference between Djkstra and A* agorihthm is that Djkstra algorihm will find the shortest path between the start node and a target node however the algorithm can be inefficient as it will waste time evaluating routes that could be ignored.

The A star algorithm travels all the edges to reach the destination from the source. So the worse case time complexity is O(E) where E is the number of edges in the graph.

Average time analysis of A star:

INPUT	Original readed graph 1 X	Twice of the graph 2 X	Tripeld of the graph 3 X
Average Time in microsecond	48.407500000000006	2705.925	24081.417499999992
Average path length	49.74999999999999	341.99749999999995	1279.14

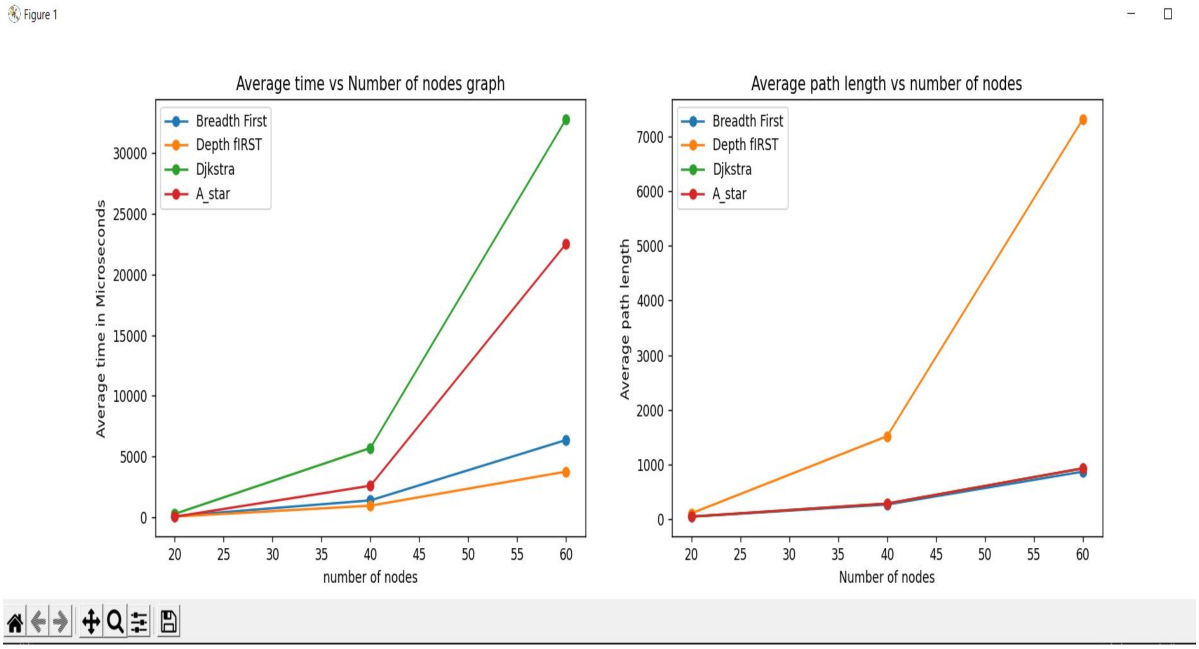


Figure 1
Average time vs Number of nodes graph
Breadth First
Depth fIRST
Djkstra
A_star
Average time in Microseconds
0
5000
10000
15000
20000
25000
30000
20
25
30
35
40
45
50
55
60
number of nodes
Average path length vs number of nodes
Breadth First
Depth fIRST
Djkstra
A_star
Average path length
0
1000
2000
3000
4000
5000
6000
7000
20
25
30
35
40
45
50
55
60
Number of nodes