

INTRODUCTION

Problem statement of the project is "Space - Efficient Algorithms for Reachability in Directed Geometric Graphs"

Using the problem statement as the base we came up with an application to find the shortest distance between a source to destination for a warehouse robot system.

The domain of the application comes under graph exploration and implementations. The applications will find the shortest path for robot so it can spend less resources reaching that destination.

Since the computers in moving robots have space and power limitations, using shortest distance technique will help us achieve better efficiency.

The goal of graph Reachability is to determine whether each graph has a path from one vertex to another.

We will aim to construct a space-efficient algorithm for reachability in a directed geometric graph in this project.

The application will determine the quickest course for the robot to take in order for it to use fewer resources to reach its objective.

Because the computers in moving robots have limited space and power, we can improve efficiency by employing the shortest distance technique.

PROBLEM STATEMENT

Problem statement – “Space - Efficient Algorithms for Reachability in Directed Geometric Graphs”

Domain – Graph exploration and implementations

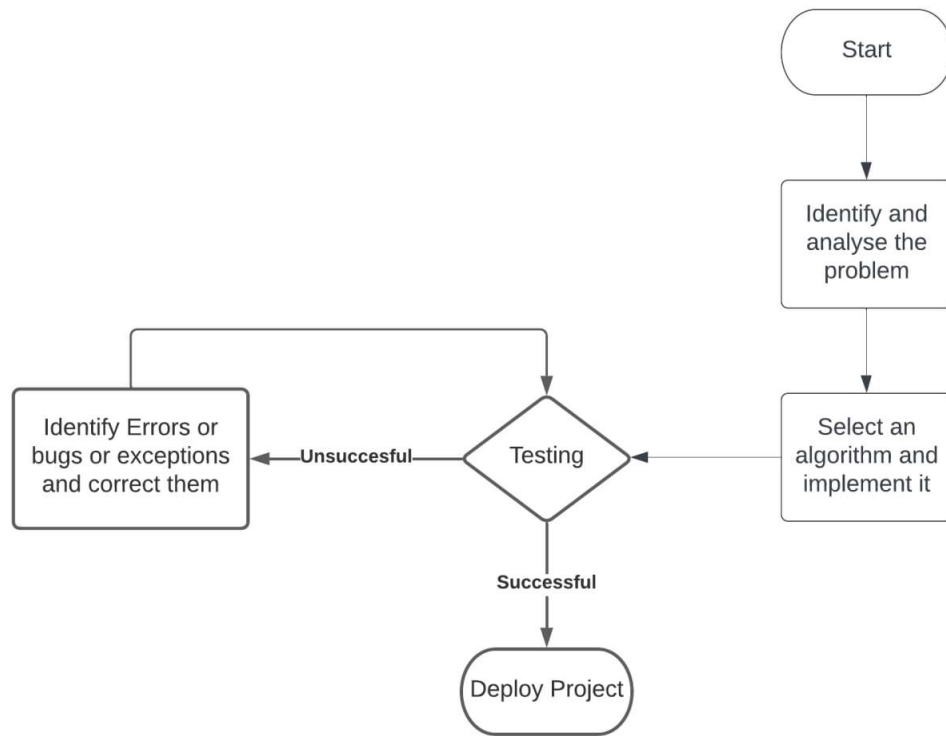
LITERATURE SURVERY

S.NO	Authors	Title	Publishing	Techniques
1	Niranka Banerjee, Sankardeep Chakraborty, Venkatesh Raman	Improved Space Efficient Algorithms for BFS, DFS and Applications	Agronomy for Sustainable Development volume (2015)	Explores the time and space complexities of BFS and DFS and experiments on the algorithms to improve the relative space complexity
2	Sujoy Bhore and Rahul Jain	Space-Efficient Algorithms for Reachability in Geometric Graphs	IEEE (2015)	The paper explains the space related optimizations while maintaining a polynomial time bound.
3	WangShu-Xi	The Improved Dijkstra's Shortest Path Algorithm and Its Application	School of Information Technology, The University of International Business and Economics, Bei Jing, China, (2017)	The shortest path problem exists in variety of areas. A well known shortest path algorithm is Dijkstra's, also called "label algorithm".
4	A Fitriansyah, N W Parwati, DR Wardhani and N Kustian	Dijkstra's Algorithm to Find Shortest Path of Tourist Destination in Bali	Journal of Physics: Conference Series, Volume 1338, The 2nd International Conference on Applied Sciences Mathematics and Informatics 9–11 August 2018, Bandar Lampung, Indonesia (2019)	Bali is one of many small island in Indonesia and referred as " <i>The Island of Gods</i> ". Bali has varieties of tourist destination e.g. art villages, ecotourism, historical building, spiritual tourism and temples. Dijkstra's algorithm is an algorithm that used to be solution in finding shortest path problem. It can be use to find the shortest route between one tourist destination and all other tourist destinations.

SOLUTION STRATEGY

- The applications will find the shortest path for robot so it can spend less resources reaching that destination.
- Proposed Technique is using Dijkstras Algorithm.
- It is easy to implement with the graph data structure and the time and space complexity is comparable with other solutions.
- Prerequisites before implementing the algorithm
 - Visiting the nodes
 - Exploration of nodes
 - Depth first search algorithm and functionality
- Time complexity – $O(\log V)$
- Space complexity - $O(V)$.
- Since the computers in a mobile robot can be limiting the problem statement objective of using the most space saving algorithm is justified.

FLOWCHART



HARDWARE AND SOFTWARE USED

Hardware –

- Processor –
 1. AMD Athlon 3000G and above
 2. Intel i3 4th gen and above
- Ram – 4Gs
- Storage - 240gs SSD / 512gb HHD

Software –

- Windows 10
- Python 3.8 and above
- visual studio code

Webhosting – flask

IMPLEMENTATION

main.py

```
import graphlib
from flask import *
import sys

app = Flask(__name__)

nodes = ["0", "1", "2", "3", "4"]

init_graph = {}
for node in nodes:
    init_graph[node] = {}

init_graph["0"]["1"] = 3
init_graph["0"]["3"] = 7
init_graph["0"]["4"] = 8
init_graph["1"]["3"] = 4
init_graph["1"]["2"] = 1
init_graph["2"]["3"] = 2
init_graph["3"]["1"] = 4
init_graph["3"]["4"] = 3

class Graph(object):
    def __init__(self, nodes, init_graph):
        self.nodes = nodes
        self.graph = self.construct_graph(nodes, init_graph)

    def construct_graph(self, nodes, init_graph):
        graph = {}
        for node in nodes:
            graph[node] = {}

        graph.update(init_graph)

        for node, edges in graph.items():
            for adjacent_node, value in edges.items():
                if graph[adjacent_node].get(node, False) == False:
                    graph[adjacent_node][node] = value

        return graph

    def get_nodes(self):
        return self.nodes
```

```

def get_outgoing_edges(self, node):
    connections = []
    for out_node in self.nodes:
        if self.graph[node].get(out_node, False) != False:
            connections.append(out_node)
    return connections

def value(self, node1, node2):
    return self.graph[node1][node2]

def dijkstra_algorithm(graph, start_node):
    unvisited_nodes = list(graph.get_nodes())
    shortest_path = {}
    previous_nodes = {}
    max_value = sys.maxsize
    for node in unvisited_nodes:
        shortest_path[node] = max_value
    shortest_path[start_node] = 0

    while unvisited_nodes:
        current_min_node = None
        for node in unvisited_nodes:
            if current_min_node == None:
                current_min_node = node
            elif shortest_path[node] < shortest_path[current_min_node]:
                current_min_node = node

        neighbors = graph.get_outgoing_edges(current_min_node)
        for neighbor in neighbors:
            tentative_value = shortest_path[current_min_node] +
graph.value(current_min_node, neighbor)
            if tentative_value < shortest_path[neighbor]:
                shortest_path[neighbor] = tentative_value
                previous_nodes[neighbor] = current_min_node

        unvisited_nodes.remove(current_min_node)

    return previous_nodes, shortest_path

def print_weight(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

    while node != start_node:
        path.append(node)
        node = previous_nodes[node]

    path.append(start_node)

```



```

    print("Min value {}".format(shortest_path[target_node]))

    return "Min value {} ".format(shortest_path[target_node])

def print_node(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

    while node != start_node:
        path.append(node)
        node = previous_nodes[node]

    path.append(start_node)

    print(" -> ".join(reversed(path)))

    return " -> ".join(reversed(path))

@app.route("/")
def index():
    return render_template("index.html");

@app.route("/index2")
def index2():
    return render_template("index2.html");

@app.route("/saveDetails", methods=["POST", "GET"])
def saveDetails():
    if request.method == "POST":
        try:
            start = request.form["startingnode"]
            desti = request.form["destinationnode"]
            print(start)
            print(desti)
            graph = Graph(nodes, init_graph)
            previous_nodes, shortest_path = dijkstra_algorithm(graph=graph,
start_node=start)
            msg = print_node(previous_nodes, shortest_path, start_node=start,
target_node=desti)
            msg1 = print_weight(previous_nodes, shortest_path,
start_node=start, target_node=desti)

        except:
            msg = "counldnt start or nodes dont exist"
        finally:
            return render_template("suc.html", msg=msg,msg1=msg1)

```

```
if __name__ == "__main__":
    app.run(debug=True)
```

Libraries –

```
import graphlib
from flask import *
import sys
```

Graph initialization –

```
nodes = ["0", "1", "2", "3", "4"]

init_graph = {}
for node in nodes:
    init_graph[node] = {}

init_graph["0"]["1"] = 3
init_graph["0"]["3"] = 7
init_graph["0"]["4"] = 8
init_graph["1"]["3"] = 4
init_graph["1"]["2"] = 1
init_graph["2"]["3"] = 2
init_graph["3"]["1"] = 4
init_graph["3"]["4"] = 3
```

Dijkstra algorithm –

```
class Graph(object):
    def __init__(self, nodes, init_graph):
        self.nodes = nodes
        self.graph = self.construct_graph(nodes, init_graph)

    def construct_graph(self, nodes, init_graph):
        graph = {}
        for node in nodes:
            graph[node] = {}

        graph.update(init_graph)

        for node, edges in graph.items():
            for adjacent_node, value in edges.items():
                if graph[adjacent_node].get(node, False) == False:
                    graph[adjacent_node][node] = value
```

```

        return graph

    def get_nodes(self):
        return self.nodes

    def get_outgoing_edges(self, node):
        connections = []
        for out_node in self.nodes:
            if self.graph[node].get(out_node, False) != False:
                connections.append(out_node)
        return connections

    def value(self, node1, node2):
        return self.graph[node1][node2]

def dijkstra_algorithm(graph, start_node):
    unvisited_nodes = list(graph.get_nodes())
    shortest_path = {}
    previous_nodes = {}
    max_value = sys.maxsize
    for node in unvisited_nodes:
        shortest_path[node] = max_value
    shortest_path[start_node] = 0

    while unvisited_nodes:
        current_min_node = None
        for node in unvisited_nodes:
            if current_min_node == None:
                current_min_node = node
            elif shortest_path[node] < shortest_path[current_min_node]:
                current_min_node = node

        neighbors = graph.get_outgoing_edges(current_min_node)
        for neighbor in neighbors:
            tentative_value = shortest_path[current_min_node] +
graph.value(current_min_node, neighbor)
            if tentative_value < shortest_path[neighbor]:
                shortest_path[neighbor] = tentative_value
                previous_nodes[neighbor] = current_min_node

        unvisited_nodes.remove(current_min_node)

    return previous_nodes, shortest_path

def print_weight(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

```

```

while node != start_node:
    path.append(node)
    node = previous_nodes[node]

path.append(start_node)

print("Min value {}".format(shortest_path[target_node]))

return "Min value {} ".format(shortest_path[target_node])

def print_node(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

    while node != start_node:
        path.append(node)
        node = previous_nodes[node]

    path.append(start_node)

    print(" -> ".join(reversed(path)))

    return " -> ".join(reversed(path))

```

printing the result functions -

```

def print_weight(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

    while node != start_node:
        path.append(node)
        node = previous_nodes[node]

    path.append(start_node)

    print("Min value {}".format(shortest_path[target_node]))

    return "Min value {} ".format(shortest_path[target_node])

def print_node(previous_nodes, shortest_path, start_node, target_node):
    path = []
    node = target_node

```

```

while node != start_node:
    path.append(node)
    node = previous_nodes[node]

path.append(start_node)

print(" -> ".join(reversed(path)))

return " -> ".join(reversed(path))

```

Flask links –

```

@app.route("/")
def index():
    return render_template("index.html");

@app.route("/index2")
def index2():
    return render_template("index2.html");

@app.route("/saveDetails", methods=["POST", "GET"])
def saveDetails():
    if request.method == "POST":
        try:
            start = request.form["startingnode"]
            desti = request.form["destinationnode"]
            print(start)
            print(desti)
            graph = Graph(nodes, init_graph)
            previous_nodes, shortest_path = dijkstra_algorithm(graph=graph,
start_node=start)
            msg = print_node(previous_nodes, shortest_path, start_node=start,
target_node=desti)
            msg1 = print_weight(previous_nodes, shortest_path,
start_node=start, target_node=desti)

            except:
                msg = "counldnt start or nodes dont exist"
            finally:
                return render_template("suc.html", msg=msg,msg1=msg1)

```

index.html –

```
<!DOCTYPE html>
```

```

<html Lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Index</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,
600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300
,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{{ url_for('static',filename='vendor/aos/aos.css')}}"
rel="stylesheet">

  <link href="{{ url_for('static',filename='vendor/aos/aos.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/bootstrap/css/bootstrap.min.css')}}"
rel="stylesheet">
  <link href="{{ url_for('static',filename='vendor/bootstrap-icons/bootstrap-
icons.css')}}" rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/boxicons/css/boxicons.min.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/glightbox/css/glightbox.min.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/remixicon/remixicon.css')}}"
rel="stylesheet">
  <link href="{{ url_for('static',filename='vendor/swiper/swiper-
bundle.min.css')}}" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="{{ url_for('static',filename='css/style.css')}}"
rel="stylesheet">

</head>

<body>

```

```

<!-- ===== Header ===== -->
<header id="header" class="fixed-top">
  <div class="container d-flex align-items-center justify-content-between">
    <h1 class="logo"><a href="index.html">S2D</a></h1>
    <!-- Uncomment below if you prefer to use an image Logo -->
    <!-- <a href="index.html" class="logo"></a>-->

    <nav id="navbar" class="navbar">
      <ul>
        <li><a class="nav-link scrollto active" href="/">Home</a></li>
        <li><a class="nav-link scrollto" href="index2">Start</a></li>
      </ul>
      <i class="bi bi-list mobile-nav-toggle"></i>
    </nav><!-- .navbar -->

  </div>
</header><!-- End Header -->

<!-- ===== Hero Section ===== -->
<section id="hero" class="d-flex align-items-center">
  <div class="container position-relative" data-aos="fade-up" data-aos-
delay="100">
    <div class="row justify-content-center">
      <div class="col-xl-7 col-lg-9 text-center">
        <h1>Source to destination application for warehouse robot
system</h1>

        </div>
      </div>
      <div class="text-center">
        <a href="index2" class="btn-get-started scrollto">Get Started</a>
      </div>

    <div id="preloader"></div>
    <a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>

    <!-- Vendor JS Files -->
    <script src="{
url_for('static',filename='vendor/purecounter/purecounter.js')}"></script>
    <script src="{ url_for('static',filename='vendor/aos/aos.js')}"></script>
    <script src="{
url_for('static',filename='vendor/bootstrap/js/bootstrap.bundle.min.js')}"></
script>
    <script src="{
url_for('static',filename='vendor/glightbox/js/glightbox.min.js')}"></script>

```

```

<script src="{ { url_for('static',filename='vendor/isotope-
layout/isotope.pkgd.min.js') }}"></script>
<script src="{ { url_for('static',filename='vendor/swiper/swiper-
bundle.min.js') }}"></script>
<script src="{ { url_for('static',filename='vendor/php-email-
form/validate.js') }}"></script>

<!-- Template Main JS File -->
<script src="{ { url_for('static',filename='js/main.js') }}"></script>

</body>

</html>

```

Index2.html –

```

<!DOCTYPE html>
<html Lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Input</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,
600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300
,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{ { url_for('static',filename='vendor/aos/aos.css') }}"
rel="stylesheet">

  <link href="{ { url_for('static',filename='vendor/aos/aos.css') }}"
rel="stylesheet">
  <link href="{ {
url_for('static',filename='vendor/bootstrap/css/bootstrap.min.css') }}"
rel="stylesheet">
  <link href="{ { url_for('static',filename='vendor/bootstrap-icons/bootstrap-
icons.css') }}" rel="stylesheet">
  <link href="{ {
url_for('static',filename='vendor/boxicons/css/boxicons.min.css') }}"
rel="stylesheet">

```



```

    <link href="{{
url_for('static',filename='vendor/glightbox/css/glightbox.min.css')}}"
rel="stylesheet">
    <link href="{{
url_for('static',filename='vendor/remixicon/remixicon.css')}}"
rel="stylesheet">
    <link href="{{ url_for('static',filename='vendor/swiper/swiper-
bundle.min.css')}}" rel="stylesheet">

    <!-- Template Main CSS File -->
    <link href="{{ url_for('static',filename='css/style.css')}}"
rel="stylesheet">

</head>

<body>

    <!-- ===== Header ===== -->
    <header id="header" class="fixed-top">
        <div class="container d-flex align-items-center justify-content-between">

            <h1 class="logo"><a href="index.html">S2D</a></h1>
            <!-- Uncomment below if you prefer to use an image Logo -->
            <!-- <a href="index.html" class="logo"></a>-->

            <nav id="navbar" class="navbar">
                <ul>
                    <li><a class="nav-link scrollto active" href="/">Home</a></li>
                    <li><a class="nav-link scrollto" href="index2">Start</a></li>
                </ul>
                <i class="bi bi-list mobile-nav-toggle"></i>
            </nav><!-- .navbar -->

        </div>
    </header><!-- End Header -->

    <!-- ===== Hero Section ===== -->
    <section id="hero" class="d-flex align-items-center">
        <div class="container position-middle" data-aos="fade-up" data-aos-
delay="100">
            <div class="row justify-content-center">
                <div class="col-xl-8 col-lg-7 text-center">
                    <form action = "./saveDetails" method="post">
                        <table>
                            <tr><td>Starting Node</td><td><input type="text"
name="startingnode"></td></tr>

```

```

        <tr><td>Destination Node </td><td><input type="text"
name="destinationnode"></td></tr>
        <tr><td><input class="btn-get-started scrollto"
type="submit" value="Submit"></td></tr>
        <img src = "{{
url_for('static',filename='img/graph.png')}}" width=400px align = "right" >

        </table>
    </form>

</div>
</div>
<div id="preloader"></div>
<a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>
<!-- Vendor JS Files -->
<script src="{{
url_for('static',filename='vendor/purecounter/purecounter.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/aos/aos.js')}}"></script>
<script src="{{
url_for('static',filename='vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></
script>
<script src="{{
url_for('static',filename='vendor/glightbox/js/glightbox.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/isotope-
layout/isotope.pkgd.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/swiper/swiper-
bundle.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/php-email-
form/validate.js')}}"></script>

<!-- Template Main JS File -->
<script src="{{ url_for('static',filename='js/main.js')}}"></script>

</body>

</html>

```

suc.html

```

<!DOCTYPE html>
<html>
<head>
    <title>suc page</title>
</head>
<body>
</body>

```

```

</html>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Output</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,
600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300
,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{{ url_for('static',filename='vendor/aos/aos.css')}}"
rel="stylesheet">

  <link href="{{ url_for('static',filename='vendor/aos/aos.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/bootstrap/css/bootstrap.min.css')}}"
rel="stylesheet">
  <link href="{{ url_for('static',filename='vendor/bootstrap-icons/bootstrap-
icons.css')}}" rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/boxicons/css/boxicons.min.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/glightbox/css/glightbox.min.css')}}"
rel="stylesheet">
  <link href="{{
url_for('static',filename='vendor/remixicon/remixicon.css')}}"
rel="stylesheet">
  <link href="{{ url_for('static',filename='vendor/swiper/swiper-
bundle.min.css')}}" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="{{ url_for('static',filename='css/style.css')}}"
rel="stylesheet">

</head>

```

```

<body>

  <!-- ===== Header ===== -->
  <header id="header" class="fixed-top">
    <div class="container d-flex align-items-center justify-content-between">

      <h1 class="logo"><a href="index.html">S2D</a></h1>
      <!-- Uncomment below if you prefer to use an image Logo -->
      <!-- <a href="index.html" class="logo"></a>-->

      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto active" href="/">Home</a></li>
          <li><a class="nav-link scrollto" href="index2">Start</a></li>
        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
      </nav><!-- .navbar -->

    </div>
  </header><!-- End Header -->

  <!-- ===== Hero Section ===== -->
  <section id="hero" class="d-flex align-items-center">
    <div class="container position-relative" data-aos="fade-up" data-aos-
delay="100">
      <div class="row justify-content-center">
        <div class="col-xl-7 col-lg-9 text-center">
          <div type="align-center">
            <h3>{{msg}}</h3>
            <h3>{{msg1}}</h3>
            <br>
            <img src = "{{ url_for('static',filename='img/graph.png')}}"
width=400px align = "center" >

          </div>
        </div>
      </div>
    </div>
    <div id="preloader"></div>
    <a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>

    <!-- Vendor JS Files -->
    <script src="{{
url_for('static',filename='vendor/purecounter/purecounter.js')}}"></script>
    <script src="{{ url_for('static',filename='vendor/aos/aos.js')}}"></script>

```

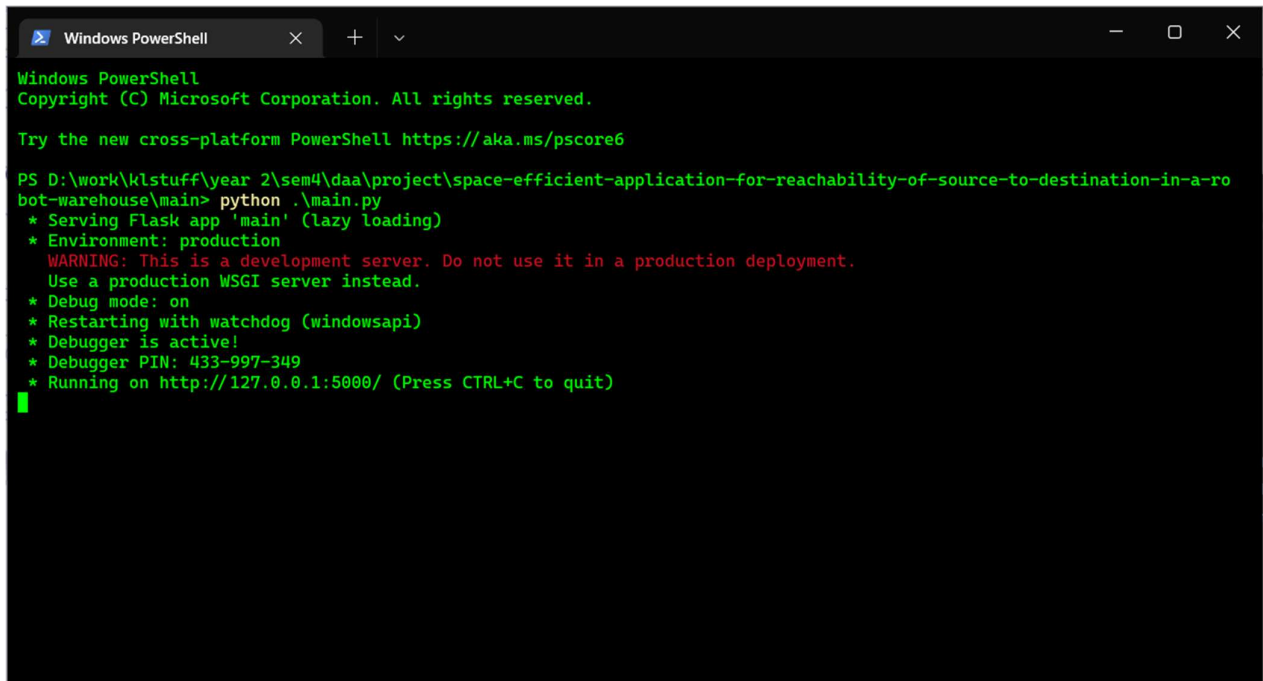
```
<script src="{{
url_for('static',filename='vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></
script>
<script src="{{
url_for('static',filename='vendor/glightbox/js/glightbox.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/isotope-
layout/isotope.pkgd.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/swiper/swiper-
bundle.min.js')}}"></script>
<script src="{{ url_for('static',filename='vendor/php-email-
form/validate.js')}}"></script>

<!-- Template Main JS File -->
<script src="{{ url_for('static',filename='js/main.js')}}"></script>

</body>

</html>
```

RESULTS

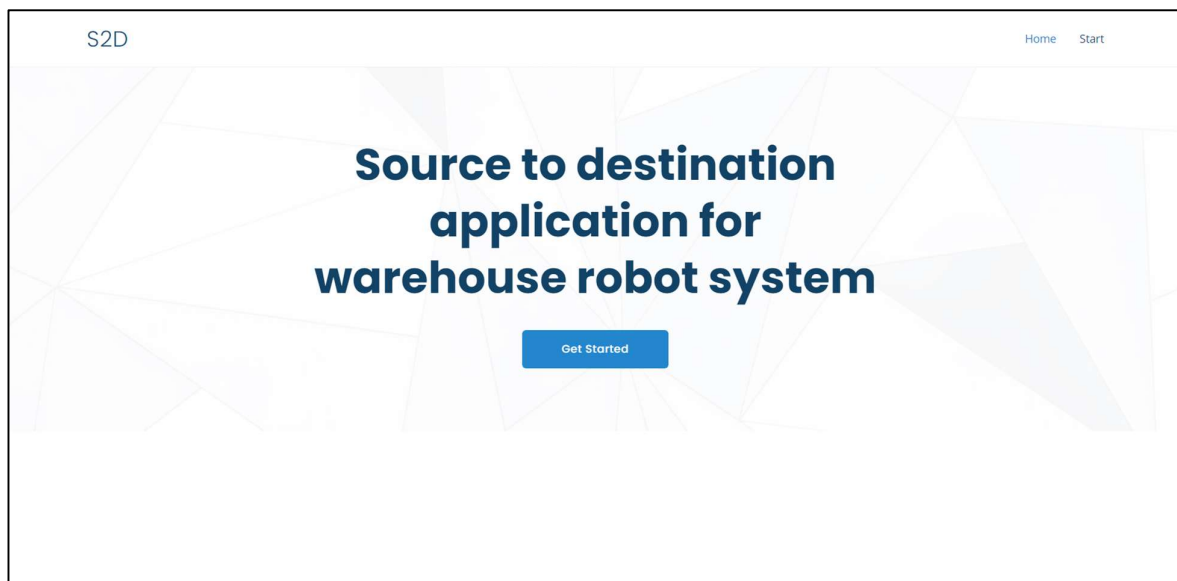


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

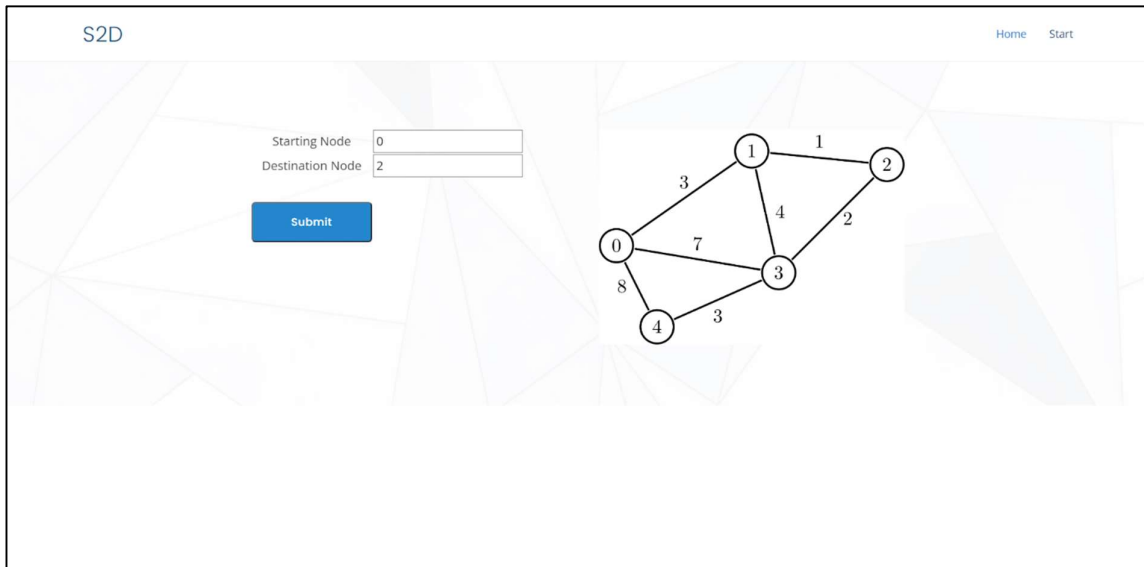
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\work\klstuff\year 2\sem4\daa\project\space-efficient-application-for-reachability-of-source-to-destination-in-a-ro
bot-warehouse\main> python .\main.py
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 433-997-349
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

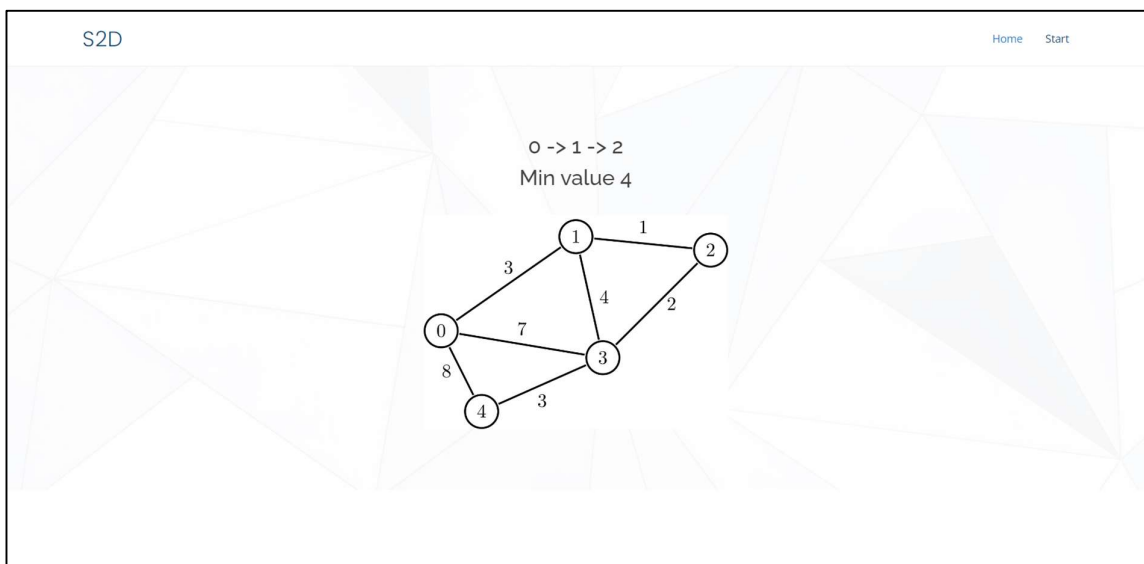
(terminal)



(Home page)



(Start/Get Started page)



(Result)

CONCLUSION & FUTURE SCOPE

- Working algorithm
- Working web interface
- Adding an ability to enter the node by the user will be an advantage to our current method
- Moving the web hosting service to Django or hosting it in a web server for testing will be appropriate representation of the intended real-life use

REFERENCES

- **Improved Space Efficient Algorithms for BFS, DFS and Applications –**
https://www.researchgate.net/publication/305455844_Improved_Space_Efficient_Algorithms_for_BFS_DFS_and_Applications
- **Space-Efficient Algorithms for Reachability in Geometric Graphs –**
<https://arxiv.org/abs/2101.05235>
- **The Improved Dijkstra's Shortest Path Algorithm and Its Application –**
<https://www.sciencedirect.com/science/article/pii/S1877705812001208>
- **Dijkstra's Algorithm to Find Shortest Path of Tourist Destination in Bali –**
<https://iopscience.iop.org/article/10.1088/1742-6596/1338/1/012044>
- **Flask -** <https://flask.palletsprojects.com/en/2.1.x/>
- **Implementing Dijkstra's Algorithm –**
<https://www.udacity.com/blog/2021/10/implementing-dijkstras-algorithm-in-python.html>
- **Dijkstra's algorithm -**
https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- **Dijkstra's algorithm -**
<https://www.programiz.com/dsa/dijkstra-algorithm>