Northeastern University
College of Engineering
Department of Electrical and Computer Engineering
EECE 2560: Fundamentals of Engineering Algorithms
Fall 2024

Project: Autonomous Robot Package Delivery

Names: Vasishta Malisetty & Jani Passas

Introduction

Objectives & Goals

- To allow delivery vehicle to most efficiently deliver packages
- To enable robot to autonomously navigate obstacles
- To develop a robust backtracking algorithm that can be applied to other fields

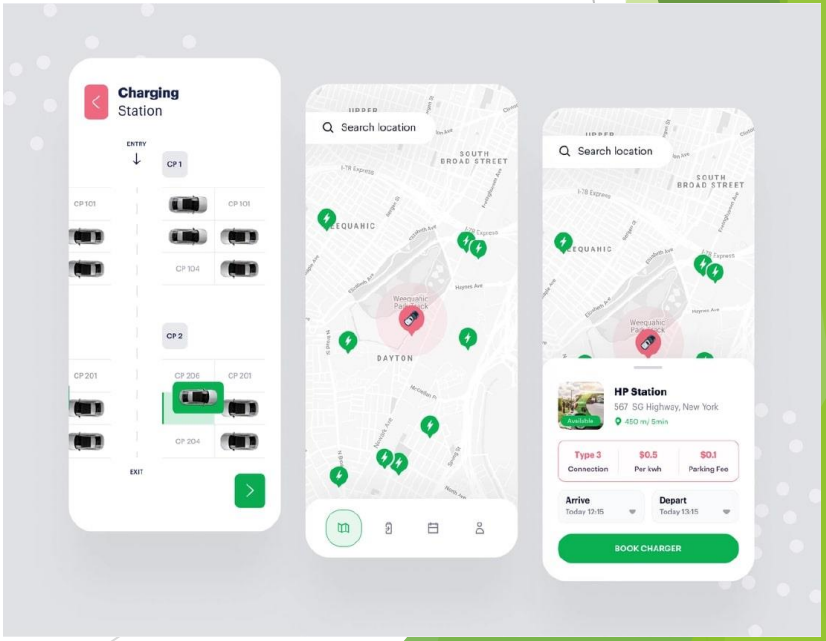
Project Scope

- Expand into greedy/graph algorithms by incorporating the car fueling along its path
- Incorporate the outputs onto a UI on a front-end web page

Literature Review (Industry)

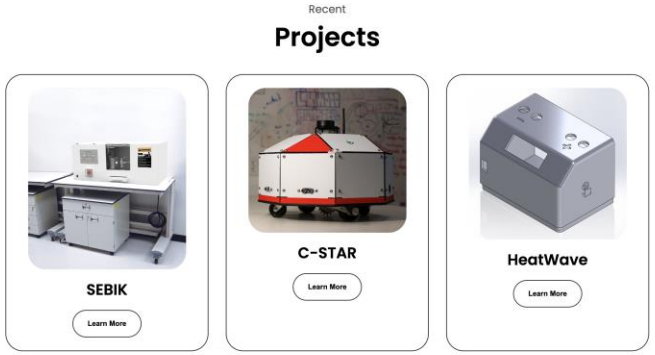
Amazon Delivery

Electric Vehicles

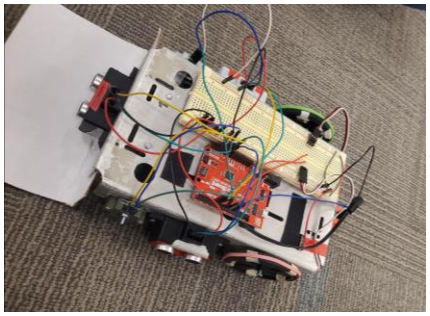


Literature Review (Personal)

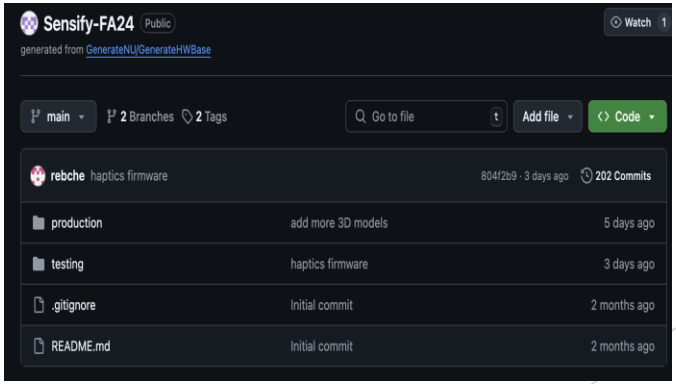
HTML & CSS

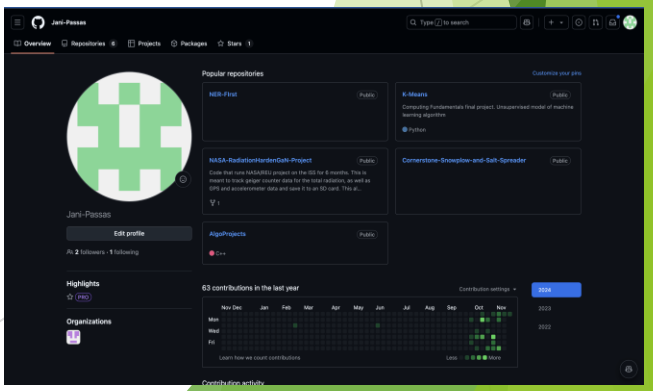


C++



Git





Data Structures

Structs

- Path
  - All paths including pathTaken
- Node
  - Weight and x, y coordinates. Used for priority queue

Vectors

- 2D matrix of maze
- Used for path tracking
- Vector pair used for directions

Queues

- Priority queue uses min-heap and greedy to always select smallest accumulated weight

Methodology (2 approaches)

Dynamic Programming

Recursively finds all paths

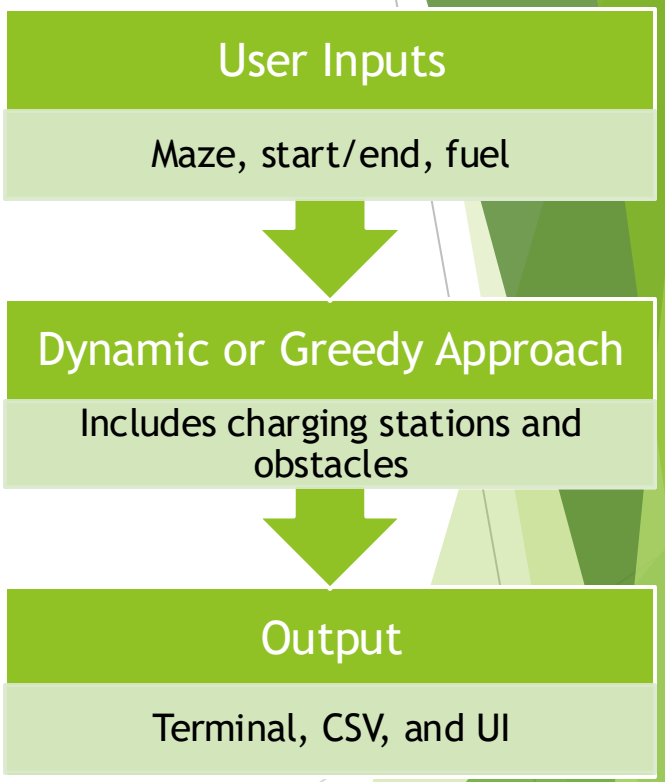
Then picks final path with minimum total weight

Greedy

Prim's MST and Dijkstra's inspired

- Use priority queue/min-heap
- Tree vs start and end

Selects smalest current cumulative weight



Back-end Overview

IsValid(x, y, maze)

- Check boundary constraints and if cell is an obstacle
- Returns true if valid, otherwise false

isValid: O(1)

HasAdjacentCharger(maze, x, y, rows, cols)

- Check possible moves (up, left, down, right)
- If neighbors are within bounds and is a charger, return true

hasAdjacentCharger: O(1)

PrintPath(finalPath, maze)

- For each cell in matrix
  If it is in path, print "1"
  If it's a charger on path or neighbor, print "C"
  If it's an unused charger, print "c"
  Else, print "0"

printPath: O(n*m)

PathToCSV(pathTaken, filename)

- Open file with filename
- Iterate through rows and write the row into the file (comma separated)
- Close the file

printPath: O(n*m)

Back-end Overview (cont)

FindAllPaths(maze, x, y, currentPath, allPaths, curWeight, endX, endY)

- If out of bounds, return
- If destination, add path to allPaths with weight
- If cell already visited, return
- For each direction
  if newCell isValid, recursively call findAllPaths to explore deeper into path until destination reached or hit dead end

findAllPaths:$O(4^{n*m})$

FindMinPath(allPaths)

- Initialize minPath
- For each path in allPaths, if path weight is less than minPath, then minPath = current path
- Return minPath

findMinPath: O(n*m)

findGreedyPath(maze, startX, startY, endX, endY, greedyPath, fuel, maxFuel, fuelExhaustion)

- Initialize visited array, parent array, usedChargers array, and priority queue
- While priority queue not empty
  - Dequeue node with smallest weight
  - If destination, trace back to start using parent array
    - Mark path in greedyPath and return total weight
  - for each direction, move and check isValid and not visisted.
    - Update weight, visit status, ad parent information.
    - Enqueue new cell into priority queue
- If destination not reached, return -1

findGreedyPath: O((n*m) log(n* m))

Log from insertion in priority queue

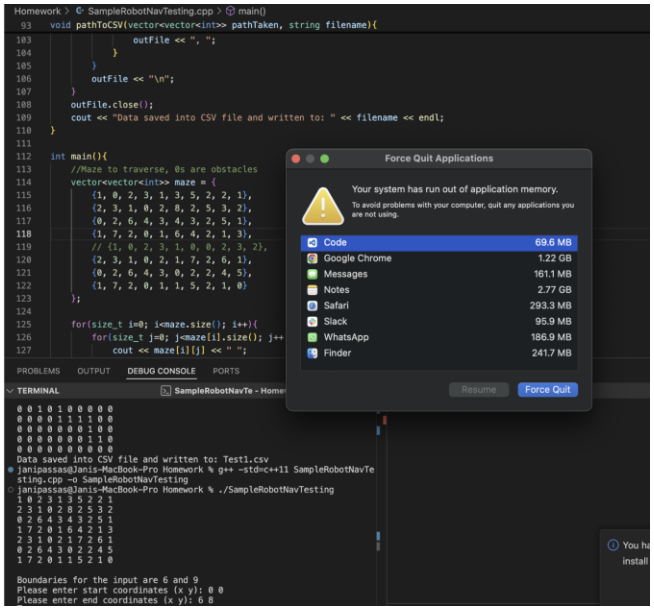
Analysis and Results (Backend)

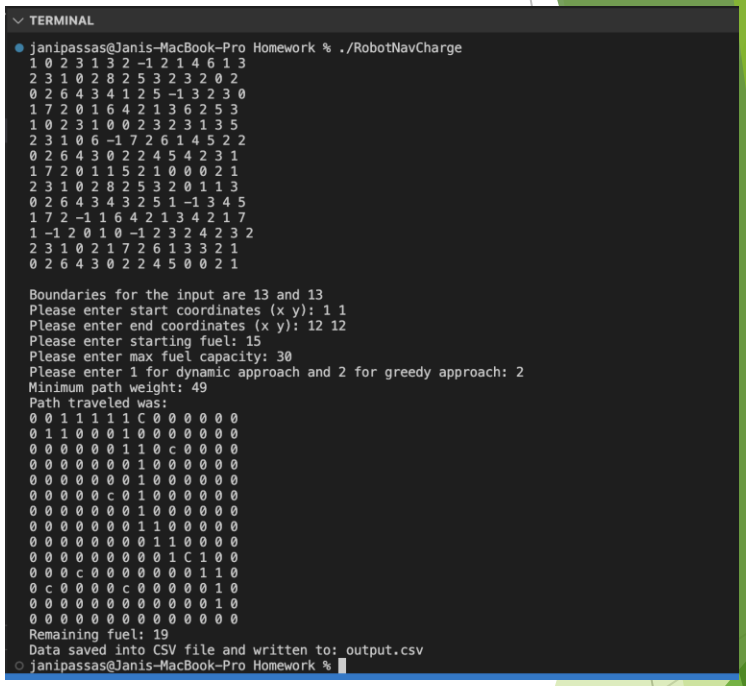
Dynamic Programming

- O(4^(n*m))

Greedy Algorithm

- O((n*m) log(n*m))



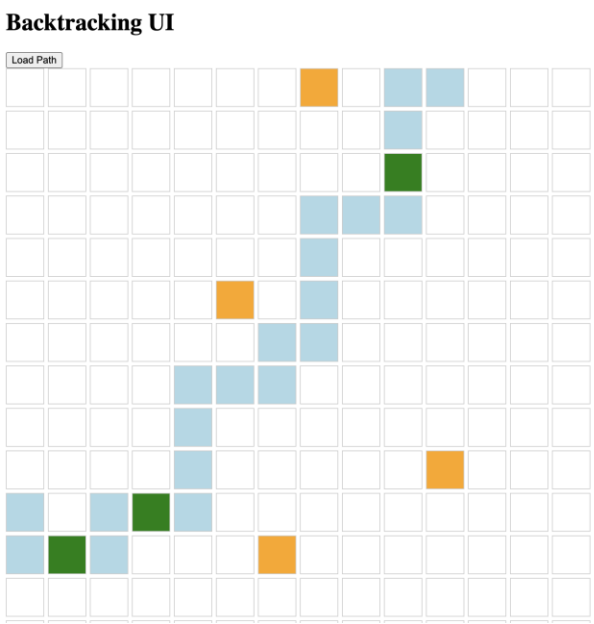


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	0	1	1	1	1	1	-2	0	0	0	0	0	0
2	0	1	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	1	0	-1	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	-1	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	-2	1	0	0
11	0	0	0	-1	0	0	0	0	0	0	0	1	1	0
12	0	-1	0	0	0	0	-1	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15														

CSV

Analysis and Results (Frontend)

UI



HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pathfinding Maze Solver</title>
    <style>
        #gridContainer {
            display: grid;
            grid-template-columns: repeat(5, 40px);
            gap: 2px;
        }
        .grid-cell {
            width: 50px;
            height: 50px;
            border: 1px solid black;
            display: flex;
            justify-content: center;
            align-items: center;
            cursor: pointer;
        }
        .path-cell {
            background-color: yellow;
        }
        .obstacle-cell {
            background-color: grey;
        }
    </style>
</head>
<body>
    <h1>Pathfinding Maze Solver</h1>
    <div id="grid" class="grid"></div>
    <button id="solveBtn">Find Shortest Path</button>
```

Server

```
vmalisetty_23@Vasishtas-MacBook-Pro AlgoProjects % python3 -m http.server

Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

Frontend Continued

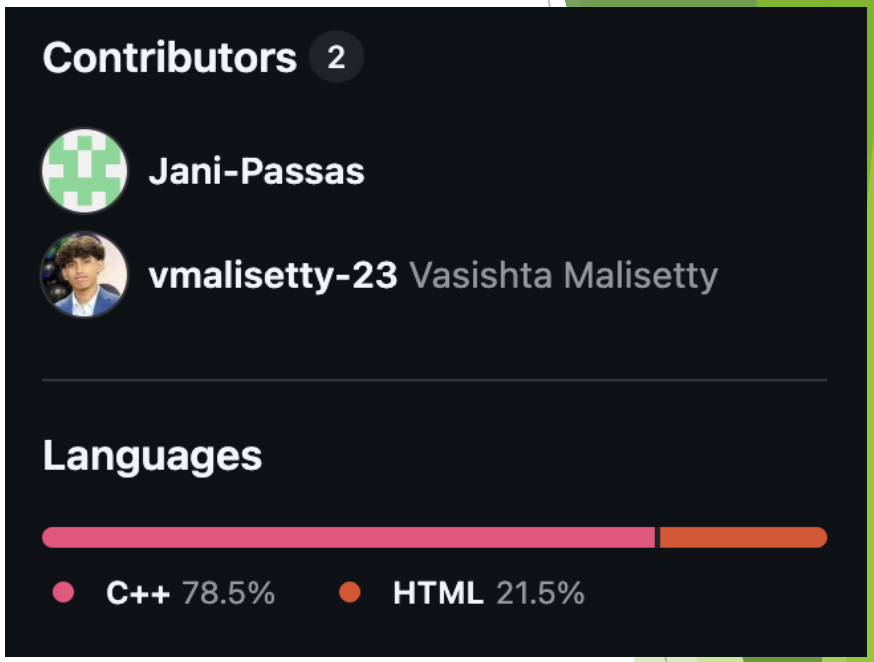
LoadCSVFile()

- Fetch CSV file
- Parse the CSV files to get rows for the UI grid
- Time Complexity: O(n)

RenderGrid()

- If grid == 1, add block to path
- Else, do not add block to the path
- Time Complexity: O(n^2)

Analysis and Results (Git)



Discussion

Limitations

- Based on an Ideal Environment
- Not currently applicable to Real-World terrain
- Memory space

Applications

- Energy Efficient Travel
- Roomba/warehouse/delivery robotics

Conclusion

Conclusions

- Greedy > Dynamic Programing
- Successfully connected Frontend & Backend

Future Steps

- Develop fully responsive UI
- Implement algorithm onto Real-World Terrain
- Add charging stations for dynamic

References

- https://www.amazon.science/latest-news/the-science-behind-grouping-amazon-package-deliveries#:~:text=delivered%20on%20time.-,Customer%20Order%20and%20Network%20Density%20OptimizeR%20(CONDOR)%20has%20led%20to,to%20identify%20effective%20delivery%20options.
- https://www.sciencedirect.com/science/article/pii/S0142061524001522#:~:text=This%20method%20refers%20to%20the,current%20research%20presented%20in%20Ref
- https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/
- https://www.geeksforgeeks.org/rat-in-a-maze/
- https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/