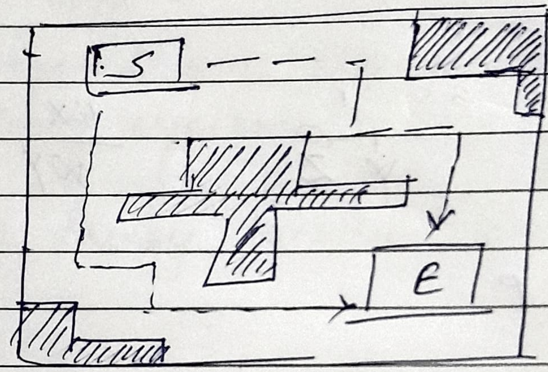
5/7/22

Motion Planning. Week - 1

- A special case of the more general planning problem.

- The goal is to develop techniques that would allow the robot to automatically decide how to move from one person to another, or from one configuration to another.

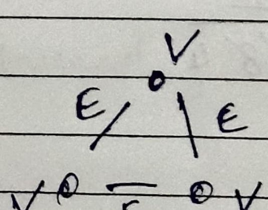


S = Start   [hatched box] - Obstacles.

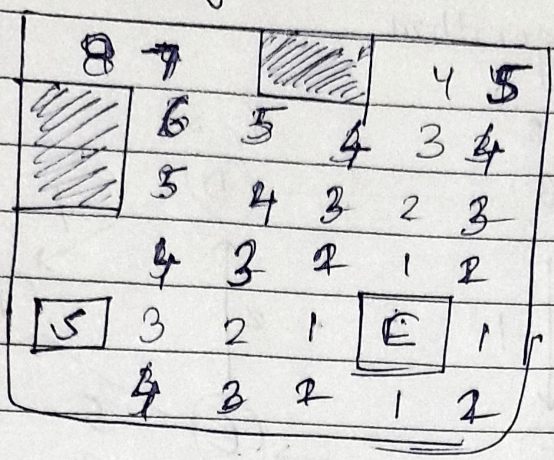
E = End

Graph Structure —

A graph G consists of set of vertices V and a set of edges E



Grassfire algorithm



Computational Complexity - Grassfire.

The computational effort required to run the grassfire algo ↑ linearly with the no. of nodes

Big-O Notation →

$O(|V|)$ $\quad$ V = no. of nodes.

No of Nodes →

2D = $100 \times 100 = 10^4$

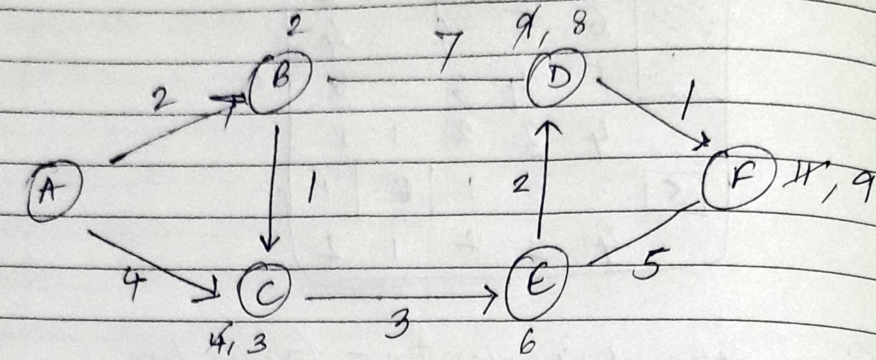
3D = $100 \times 100 \times 100 = 10^6$

6D = 10^{12}

5

[Single Source Shortest Path]

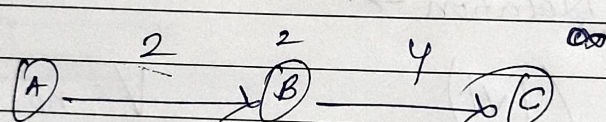
Dijkstra's Algorithm



Relaxation —

$$\text{if } \left(d[u] + c(u,v) < d[v]\right)$$

$$d[v] = d[u] + c(u,v)$$



$d[u] = 2$ $\quad d[v] = \infty$

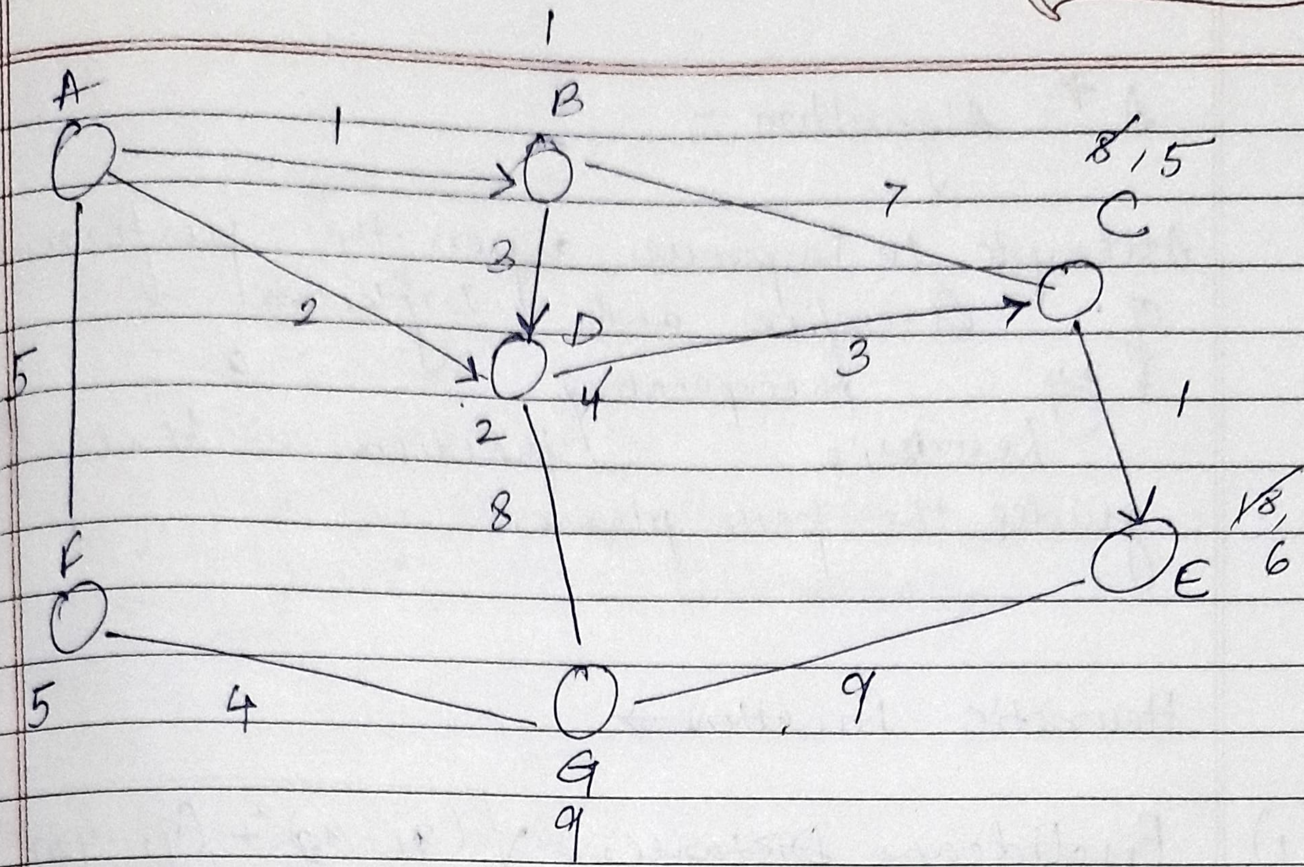
$c(u,v) = 4$

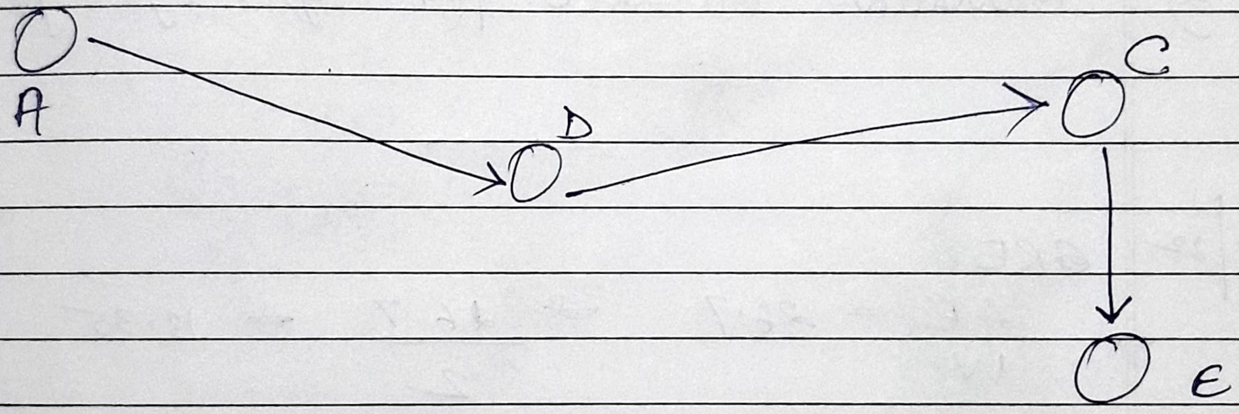
$\therefore d[v] = 2 + 4 = 6$

Complexity $\rightarrow O(n^2)$

Limitations —

- Dijkstra's algo may work or may not work if we have negative edges.





By using priority queue →

Complexity → $O((|E|+|V|)\log(|V|))$

Makes the algo more effiecent.

A^* Algorithm –

Attempts to improve upon the performance of Grassfire and Dijkstra by incorporating a heuristic function that guides the path planner.

Heuristic functions →

1) Euclidean Distance $\sqrt{(x_1 - x_g)^2 + (y_1 - y_g)^2}$

2) Manhattan Distance. $|(x_1 - x_g) + (y_1 - y_g)|$

classmate

Date ______
Page ______

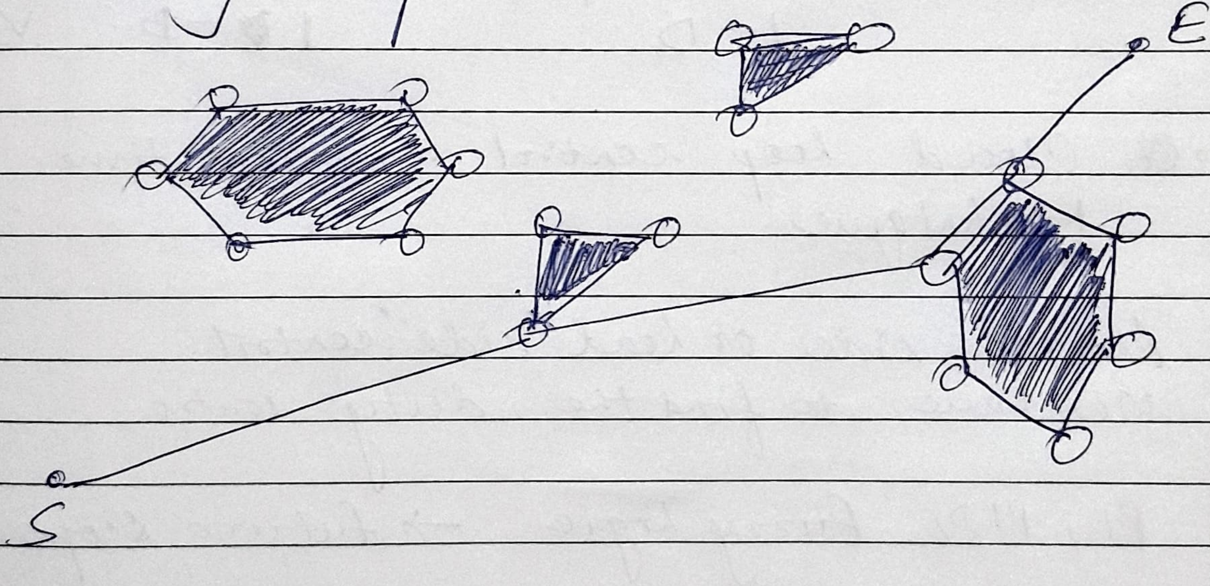
1/7/22 A* Algorithm →

Quiz-1

a) Same length – Yes
b) same except opposite – NO
c) Dimensione ↑ – Exponentially.
d) A*

11/7/22 Path planning in configuration space.

Visibility Graph –



Week - 3

5 | 12/7/22

Probablistic Road Maps (PRM)

Random Graph Construction.

- Configuration space model works well for 2, 3D but not more.

PRM Procedure —

- Repeat n times.

1. Generate a random points for configure space, x.

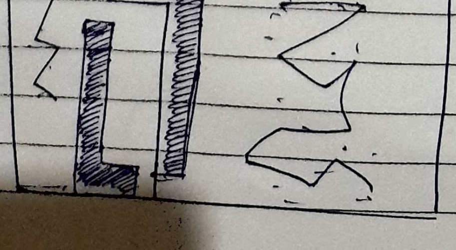
2. If x is a free space:

a) Find the k closest points in the roadmap to x according to the Dist function.

b) Try to connect the new random sample to each of the k neighbors using the local planner procedure. Each successful connections forms a new edge in the graph.

Drawbacks —

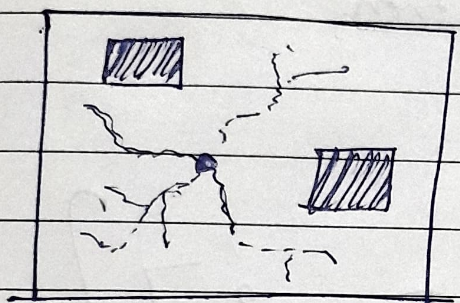
With the PRM procedure it may fail to find a path even if one exists.



It fails to find a route through the narrow passage

PRM can be applied to samples with a lot of DOFs as opposed to grid based sampling schemes.

Rapidly exploring Random Tree (RRT) Method.



A visualization of how RRT method works.

RRT is a very effective path finding algorithm

2 tree procedure where we have one tree at the start (A) and one tree at the end location or goal. (B)

We then start the RRT planner from both ends till they meet. And that will give us the path from A to B.

Constructing Artificial Potential Fields

Create a smooth function over the extent of the configuration space which has high values near obstacles and vice versa.

Lowest value at goal location

Current position →

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_g = \begin{bmatrix} x_1^g \\ x_2^g \end{bmatrix}$$

$$f_a(x) = \xi \| x - x_g \|^2$$

Attractive potential function

ξ constant scaling parameter.

Repulsive Potential Field

A repulsive potential function in the plane $f_r(x)$ can be constructed.

As it approaches an obstacles it's value increases and vice versa.

Problem with this method

- The problem of getting stuck at a local minimum.