



----- Artificial Intelligence Task -----

Route Finding Algorithms



CONTENTS

01

PART 01

Explain the idea of the application.

02

PART 02

Explain how to show the cost between nodes.

03

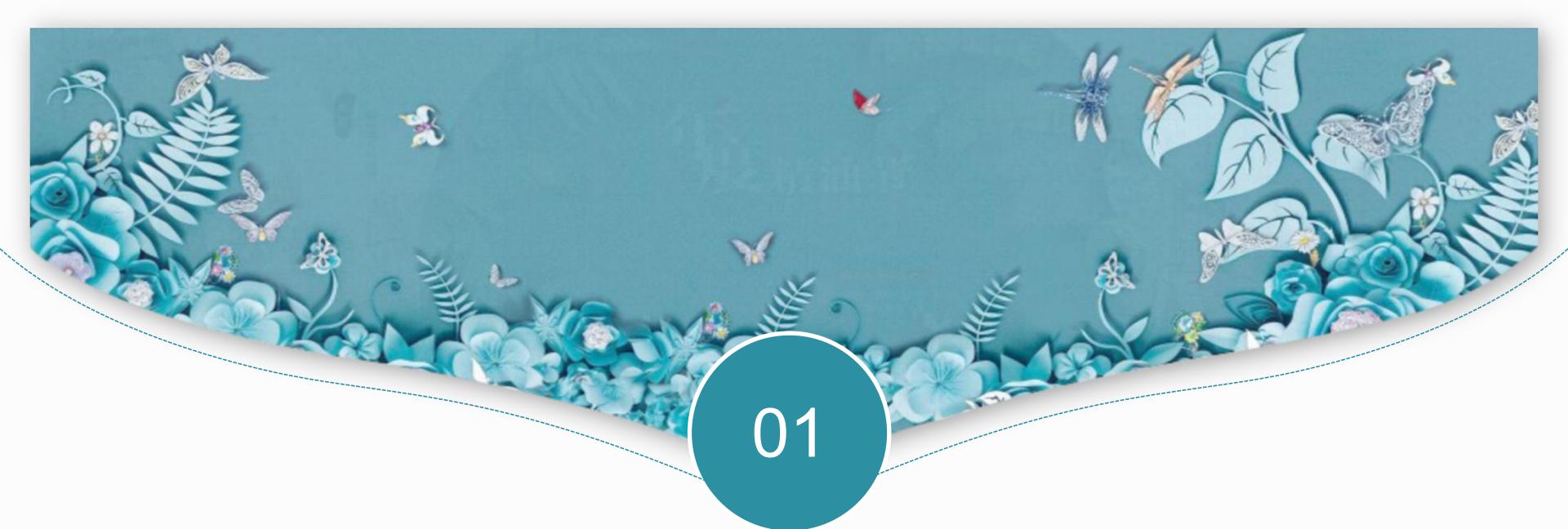
PART 03

Explain UCS Algorithm.

04

PART 04

Explain A* Algorithm.



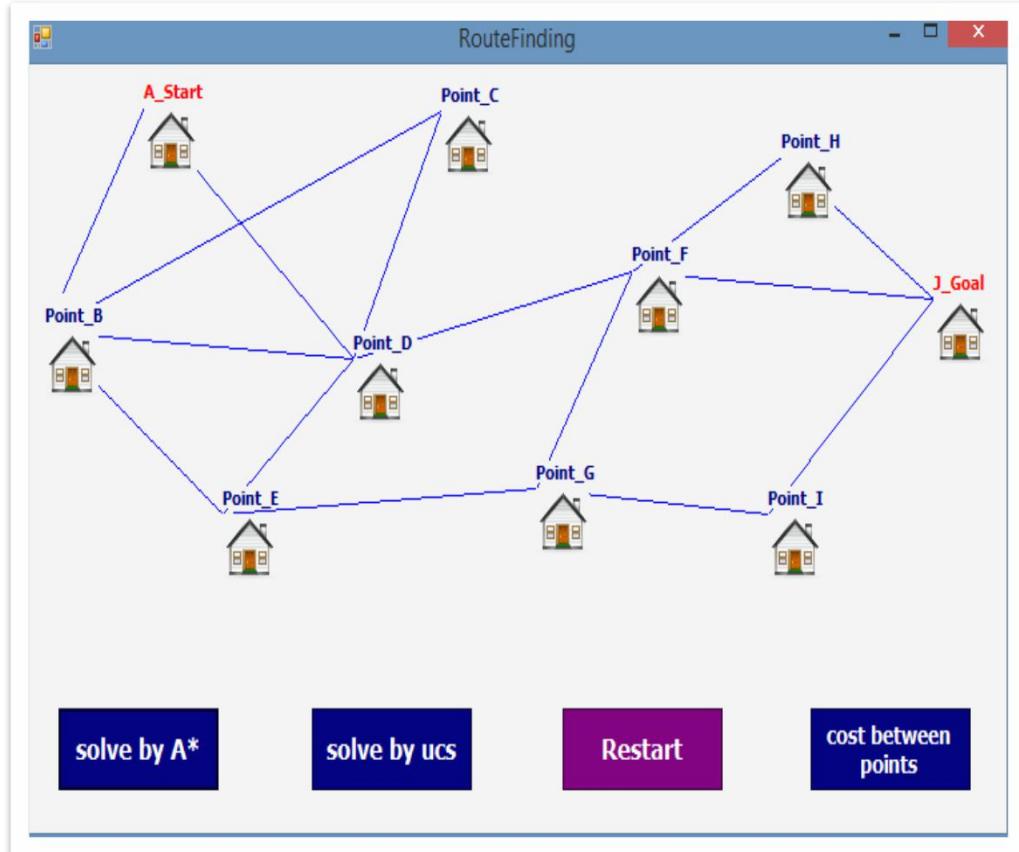
01

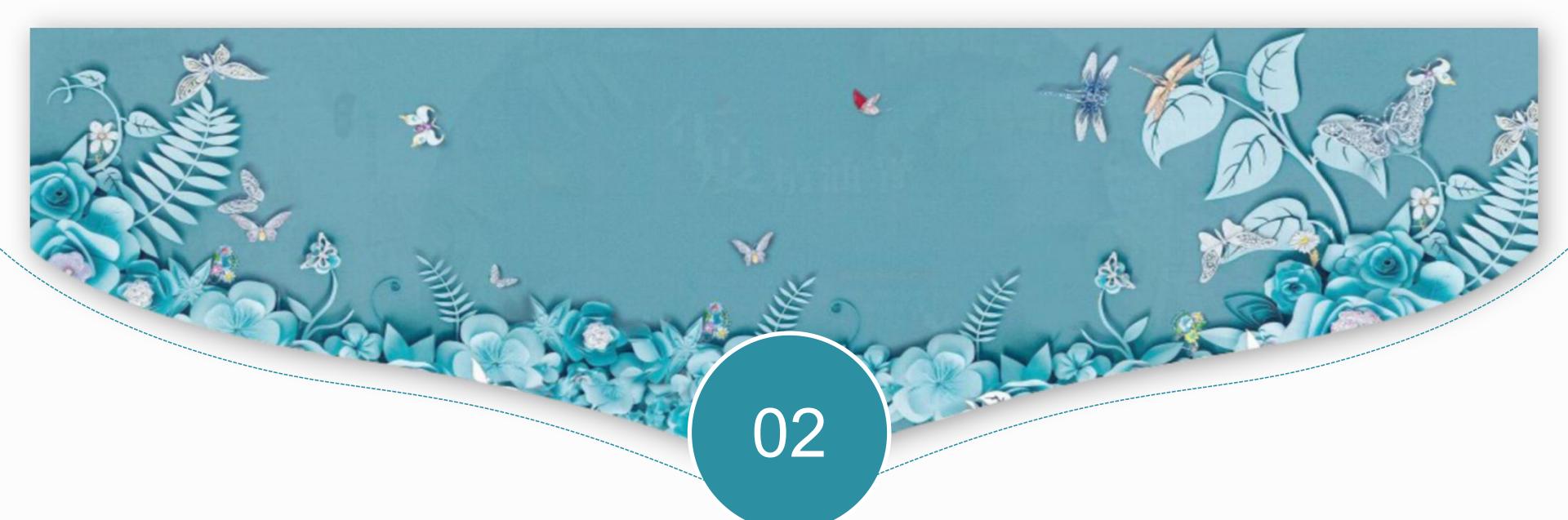
The idea of application

Idea of app

Here we explain our app idea. The game contains a group of nodes connected together in a number of ways. The required is choosing the shortest path to reach the specified node with the lowest cost.

To find the shortest path, our app uses one of two Algorithms : [UCS](#) or [A*](#) algorithms.





02

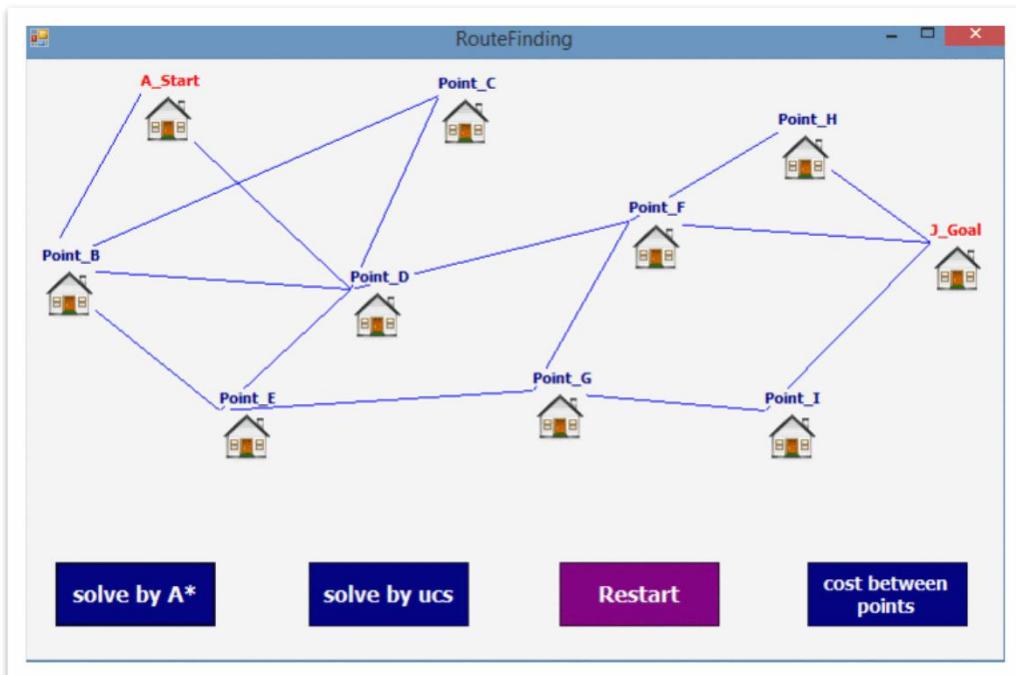
Cost between nodes

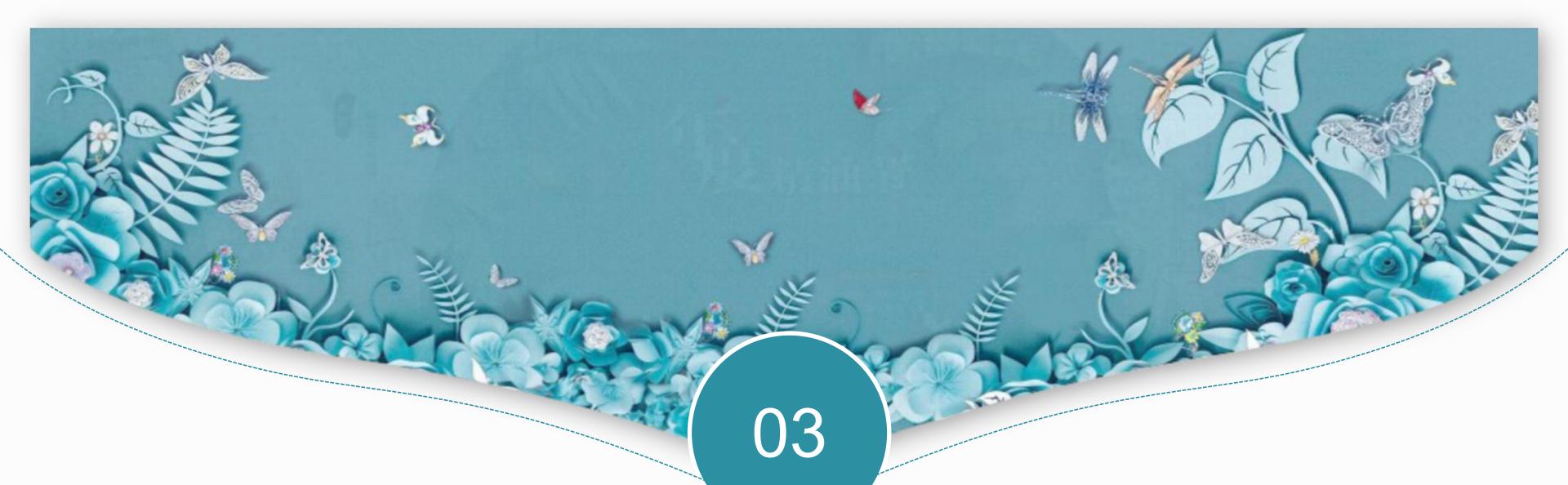
Cost between nodes

In order to see the cost between each two nodes, press on the button that name is cost between points.

A small screen will appear, it showing the cost between each two nodes in application.

To exit from this screen, click the ok button or exit mark.

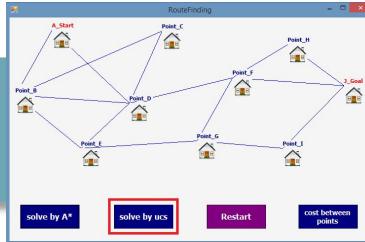




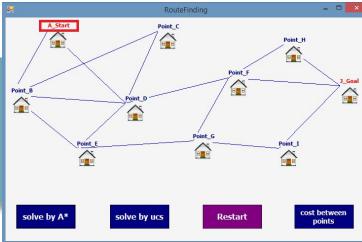
03

UCS Algorithm

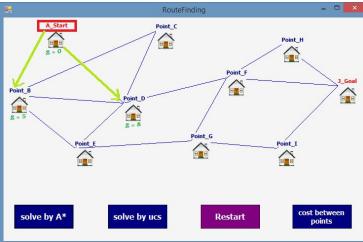
Find shortest path by using UCS algorithm



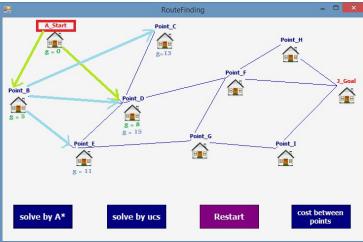
click button that solve by ucs algorithm.



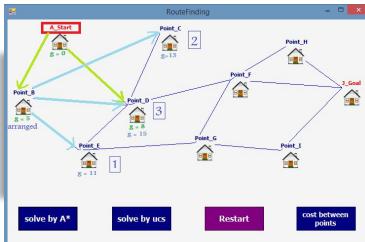
it start from first node.



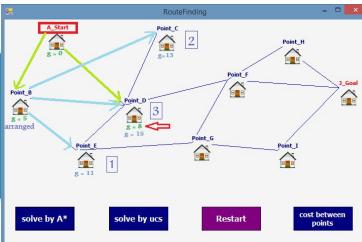
determine the nodes that connected with start node.



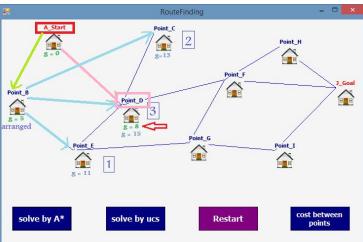
determine the nodes that connected with B node.



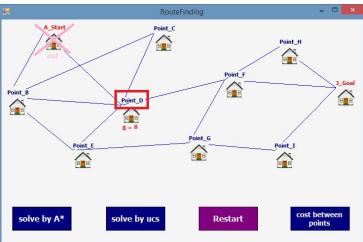
arrange nodes that connected with B node from smallest to biggest.



from A to D in direct is smallest path than from A to B and to D.

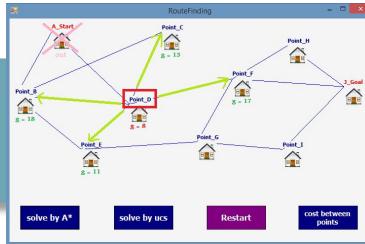


determine the next node , that will be there in next step.

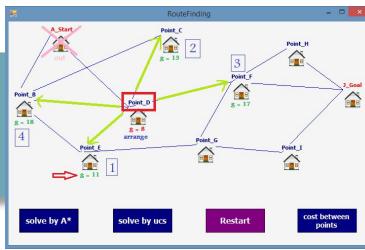


it now in D node.

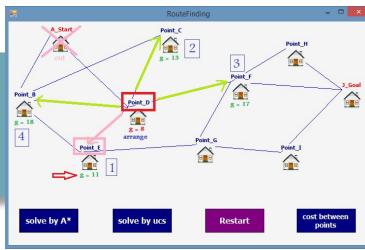
Find shortest path by using UCS algorithm



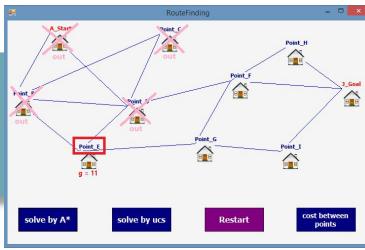
determine the nodes that connected with D node.



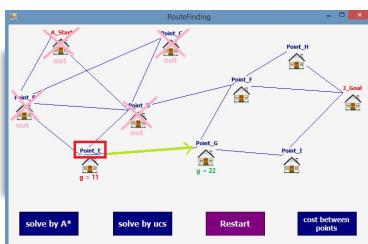
determine the smallest node.



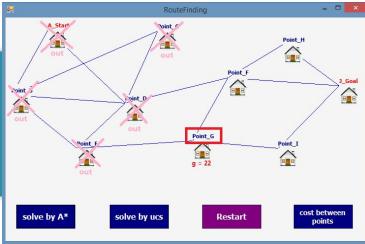
determine the next node , that will be there in next step.



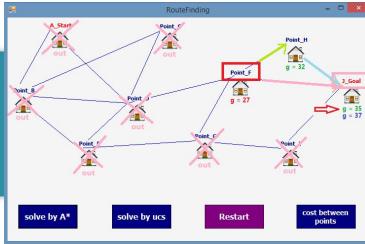
it now in E node.



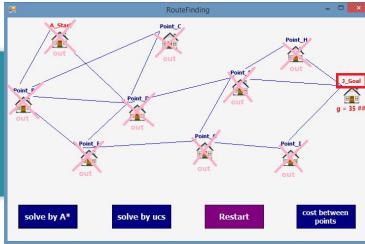
determine the nodes that connected with E node.



it now in G node.

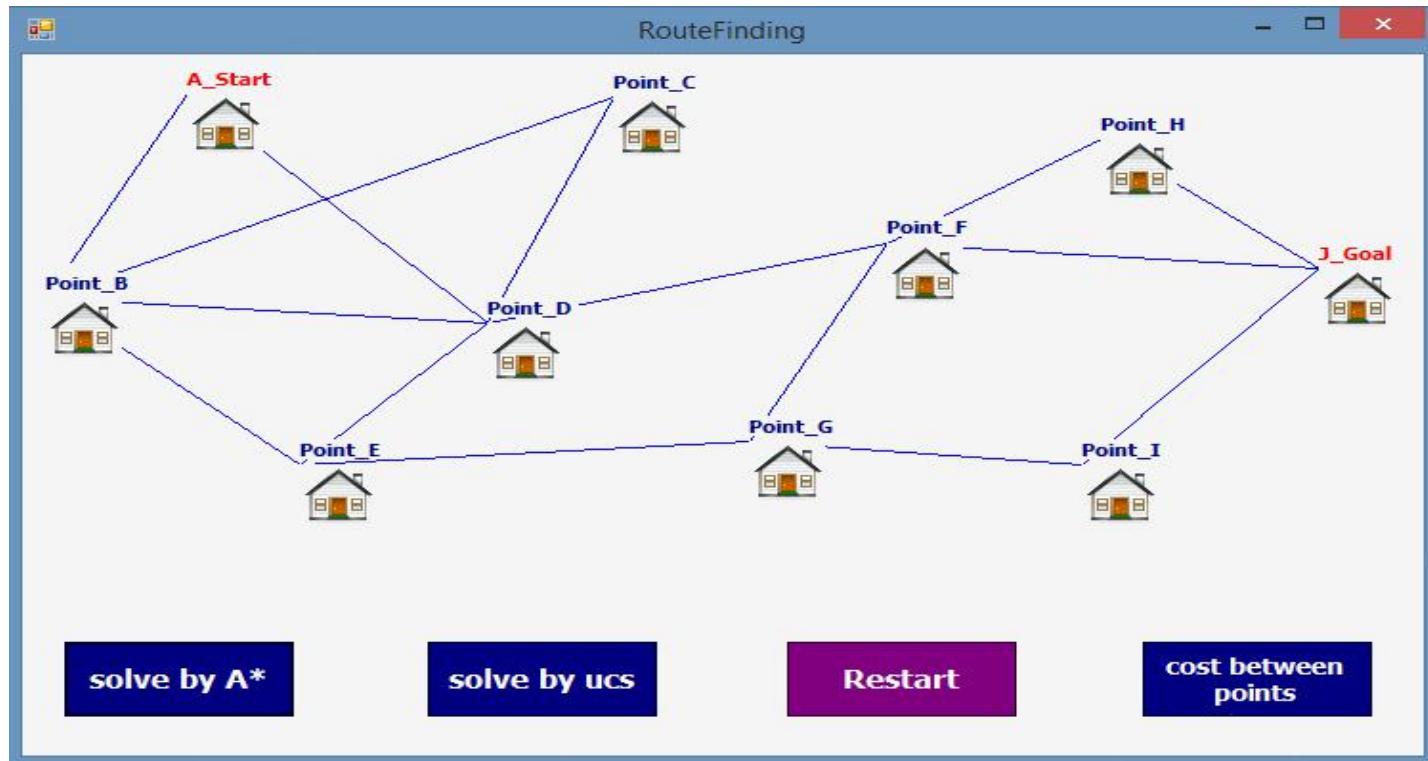


it will use algorithme until arrive to goal.

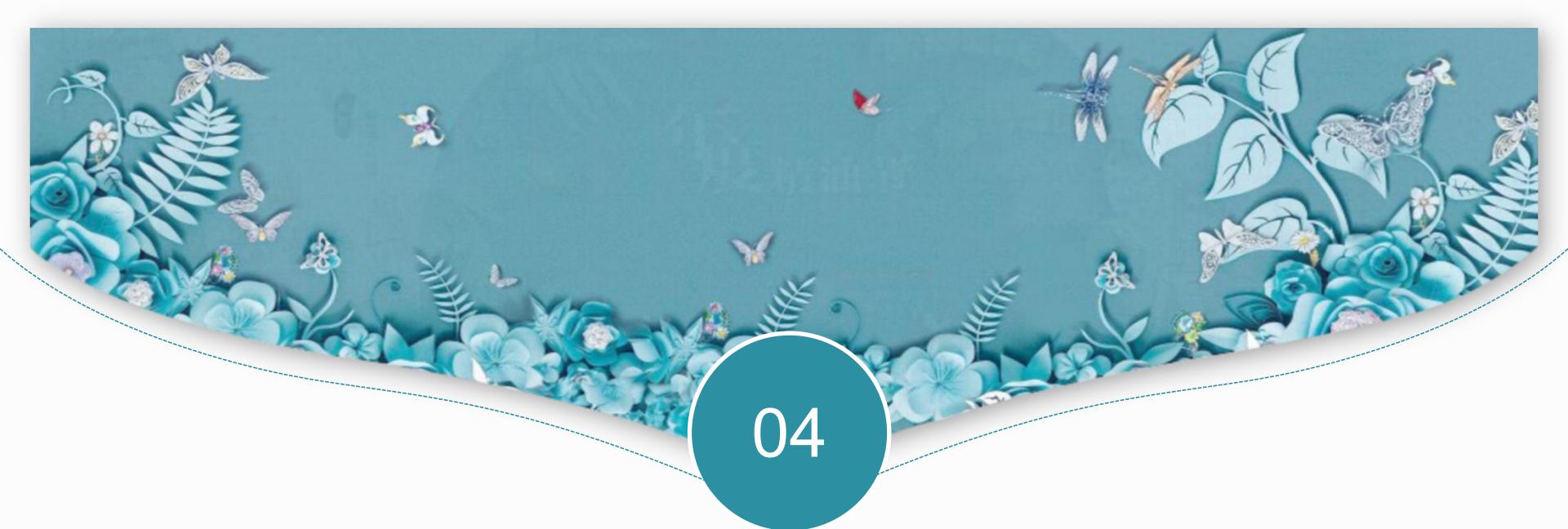


now it is in goal node.

Find shortest path by using UCS algorithm



GIF image that explain ucs algorithm in our app



04

A* Algorithm

Find shortest path by using A* algorithm

A* Algorithm

- A* Algorithm is one of the best and popular techniques used for path finding and graph traversals.

Working

- A* Algorithm extends the path that minimize the following function : $f(n) = g(n) + h(n)$

==>Here

- 'n' is the last node on the path
- $g(n)$ is the cost of the path from start node to node 'n'
- $h(n)$ is a heuristic function that estimates cost of the cheapest path from node 'n' to the goal node

Find shortest path by using A* algorithm

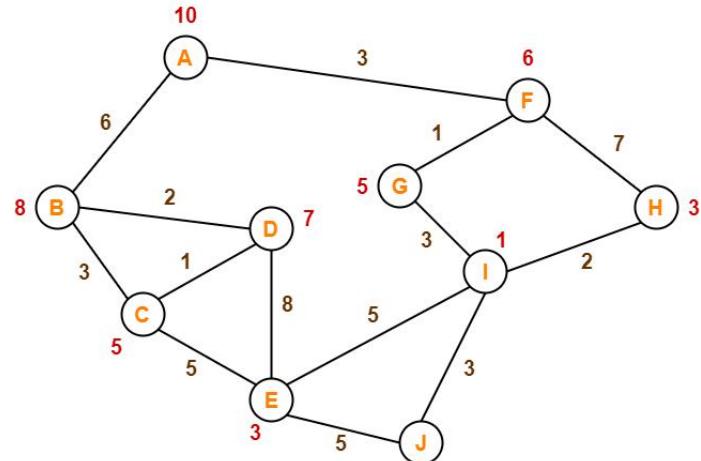
Ex :

Consider the following graph -

The numbers written on edges represent the distance between the nodes.

The numbers written on nodes represent the heuristic value.

Find the most cost-effective path to reach from start state A to final state J using A* Algorithm.



Find shortest path by using A* algorithm

Solution :

step-01:

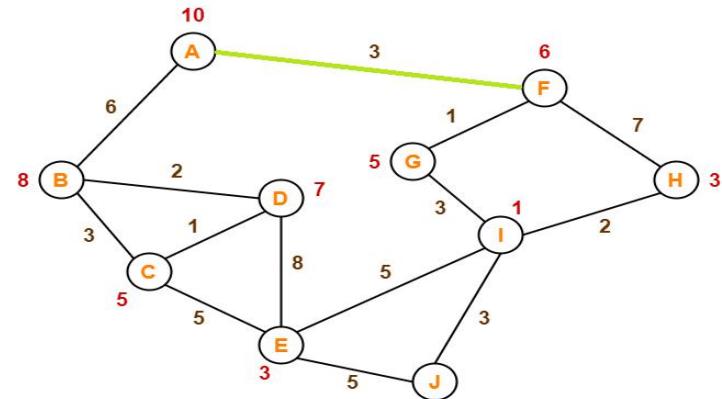
- we start with node A.
- Node B and Node F can be reached from node A.

A* Algorith calculates $f(B)$ and $f(F)$.

- $f(B) = 6 + 8 = 14$
- $f(F) = 3 + 6 = 9$

Since $f(F) < f(B)$, so it decides to go to node F.

Path : A → F



Find shortest path by using A* algorithm

Solution :

step-02:

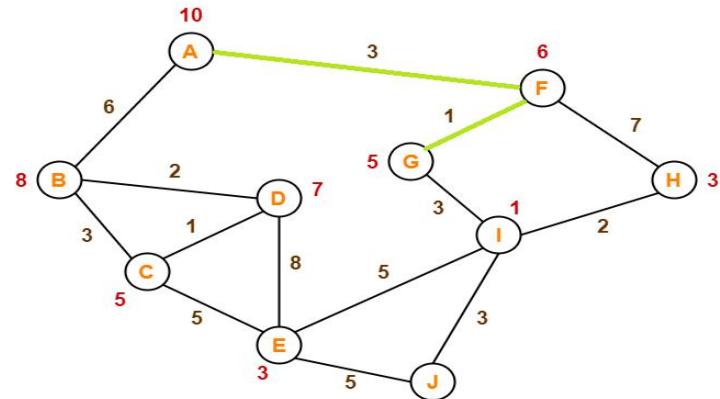
Node G and Node H can be reached from node F.

A* Algorithm calculates $f(G)$ and $f(H)$.

- $f(G) = (3+1) + 5 = 9$
- $f(H) = (3+7) + 3 = 13$

Since $f(G) < f(H)$, so it decides to go to node G.

Path : A → F → G



Find shortest path by using A* algorithm

Solution :

step-03:

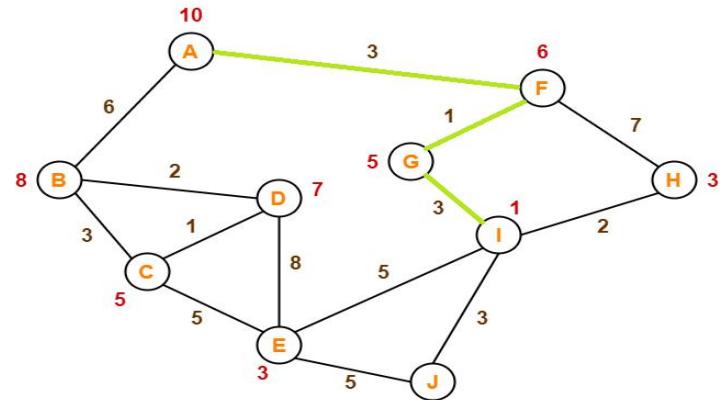
Node I can be reached from node G.

A* Algorithm calculates $f(I)$.

$$f(I) = (3+1+3) + 1 = 8$$

It decides to go to node I.

Path : A → F → G → I



Find shortest path by using A* algorithm

Solution :

step-04:

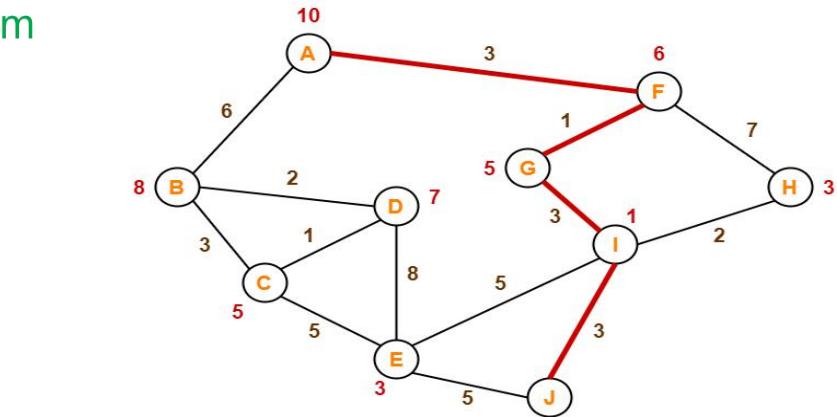
Node E, Node H and Node J can be reached from node I.

A* Algorithm calculates $f(E)$, $f(H)$ and $f(J)$.

- $f(E) = (3+1+3+5) + 3 = 15$
- $f(H) = (3+1+3+2) + 3 = 12$
- $f(J) = (3+1+3+3) + 0 = 10$

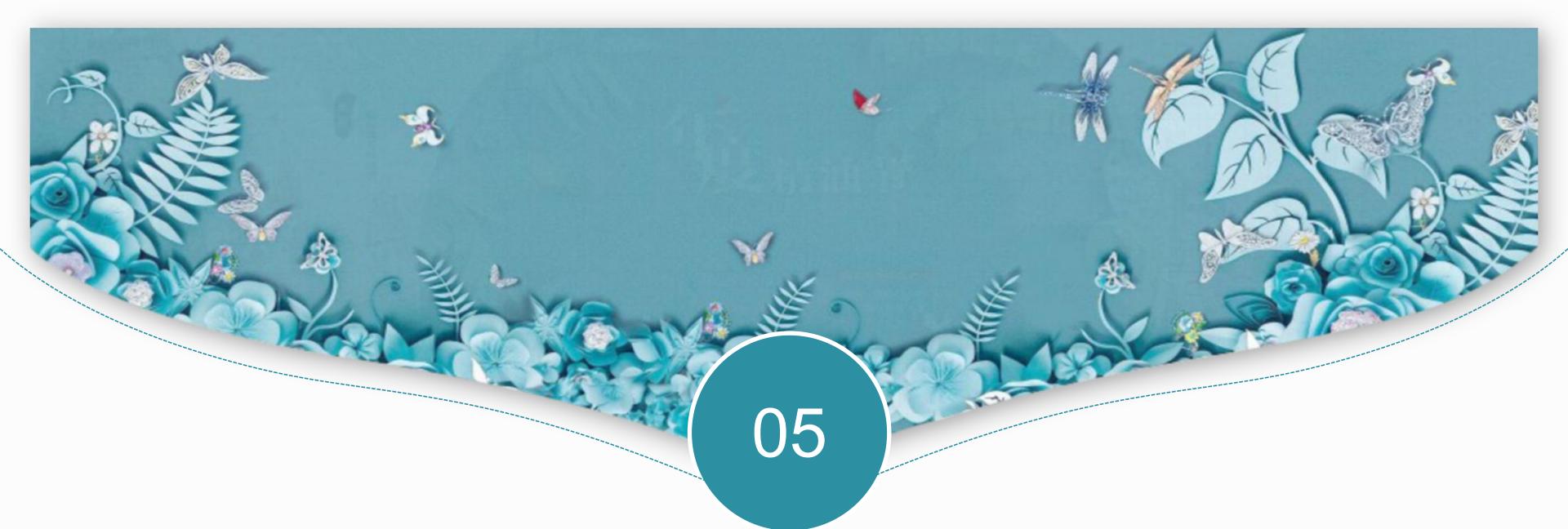
Since $f(J)$ is least, so it decides to go to node J.

Path : A → F → G → I → J



This is the required shortest path from node A to node J.

A* reference : <https://www.gatevidyalay.com/a-algorithm-a-algorithm-example-in-ai/>



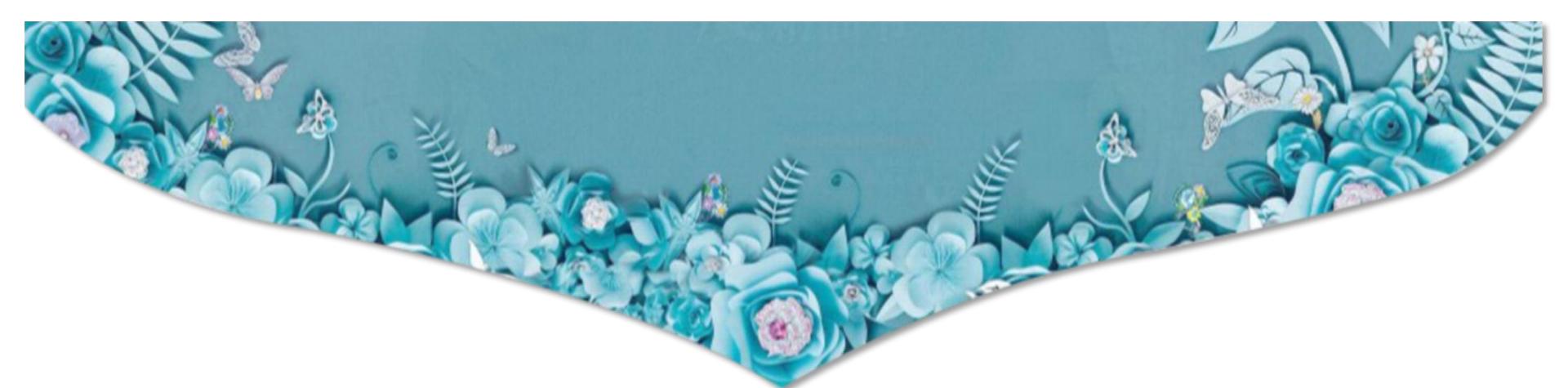
05

our team ^_^

Names of team members

all members from sec 2





THANKS