JDOODLE

Online Java Compiler IDE

For Multiple Files, Custom Library and File Read/Write, use our new - Advanced Java IDE

```
import java.util.Comparator;
import java.util.InputMismatchException;
import java.util.PriorityQueue;
import java.util.Scanner;

public class BestFirstSearch
{
    private PriorityQueue<Vertex> priorityQueue;
    private int heuristicvalues[];
    private int numberOfNodes;

    public static final int MAX_VALUE = 999;

    public BestFirstSearch(int numberOfNodes)
    {
        this.numberOfNodes = numberOfNodes;
        this.priorityQueue = new PriorityQueue<Vertex>(this.numberOfNodes,
        new Vertex());
    }

    public void bestFirstSearch(int adjacencyMatrix[][], int[] heuristicvalues,int source)
    {
        int evaluationNode;
        int destinationNode;
        int visited[] = new int [numberOfNodes + 1];
        this.heuristicvalues = heuristicvalues;

        priorityQueue.add(new Vertex(source, this.heuristicvalues[source]));
        visited[source] = 1;

        while (!priorityQueue.isEmpty())
        {
            evaluationNode = getNodeWithMinimumHeuristicValue();
            destinationNode = 1;

            System.out.print(evaluationNode + "\t");
            while (destinationNode <= numberOfNodes)
            {
                Vertex vertex = new Vertex(destinationNode,this.heuristicvalues[destinationN
                if ((adjacencyMatrix[evaluationNode][destinationNode] != MAX_VALUE
                      && evaluationNode != destinationNode)&& visited[destinationNode] == 0)
                {
                    priorityQueue.add(vertex);
                    visited[destinationNode] = 1;
                }
                destinationNode++;
            }
        }
    }

    private int getNodeWithMinimumHeuristicValue()
    {
        Vertex vertex = priorityQueue.remove();
        return vertex.node;
    }

    public static void main(String... arg)
    {
        int adjacency_matrix[][];
        int number_of_vertices;
```

```java
        int source = 0;
        int heuristicvalues[];

        Scanner scan = new Scanner(System.in);
        try
        {
            System.out.println("Enter the number of vertices");
            number_of_vertices = scan.nextInt();
            adjacency_matrix = new int[number_of_vertices + 1][number_of_vertices + 1];
            heuristicvalues = new int[number_of_vertices + 1];

            System.out.println("Enter the Weighted Matrix for the graph");
            for (int i = 1; i <= number_of_vertices; i++)
            {
                for (int j = 1; j <= number_of_vertices; j++)
                {
                    adjacency_matrix[i][j] = scan.nextInt();
                    if (i == j)
                    {
                        adjacency_matrix[i][j] = 0;
                        continue;
                    }
                    if (adjacency_matrix[i][j] == 0)
                    {
                        adjacency_matrix[i][j] = MAX_VALUE;
                    }
                }
            }
            for (int i = 1; i <= number_of_vertices; i++)
            {
                for (int j = 1; j <= number_of_vertices; j++)
                {
                    if (adjacency_matrix[i][j] == 1 && adjacency_matrix[j][i] == 0)
                    {
                        adjacency_matrix[j][i] = 1;
                    }
                }
            }

            System.out.println("Enter the heuristic values of the nodes");
            for (int vertex = 1; vertex <= number_of_vertices; vertex++)
            {
                System.out.print(vertex + ".");
                heuristicvalues[vertex] = scan.nextInt();
                System.out.println();
            }

            System.out.println("Enter the source ");
            source = scan.nextInt();

            System.out.println("The graph is explored as follows");
            BestFirstSearch bestFirstSearch = new BestFirstSearch(number_of_vertices);
            bestFirstSearch.bestFirstSearch(adjacency_matrix, heuristicvalues,source);

       } catch (InputMismatchException inputMismatch)
       {
           System.out.println("Wrong Input Format");
       }
       scan.close();
   }
}

class Vertex implements Comparator<Vertex>
{
    public int heuristicvalue;
    public int node;

    public Vertex(int node, int heuristicvalue)
    {
        this.heuristicvalue = heuristicvalue;
        this.heuristicvalue = heuristicvalue;
        this.node = node;
    }

    public Vertex()
    {

    }

    @Override
    public int compare(Vertex vertex1, Vertex vertex2)
    {
        if (vertex1.heuristicvalue < vertex2.heuristicvalue)
            return -1;
        if (vertex1.heuristicvalue > vertex2.heuristicvalue)
            return 1;
        return 0;
    }

    @Override
    public boolean equals(Object obj)
    {
        if (obj instanceof Vertex)
        {
            Vertex node = (Vertex) obj;
            if (this.node == node.node)
            {
                return true;
            }
        }
        return false;
    }
}

```

Execute Mode, Version, Inputs & Arguments

CommandLine Arguments

Result

compiled and executed in 55.895 sec(s)

```
Enter the number of vertices
5
Enter the Weighted Matrix for the graph
1 0 0 0 0
0 0 0 0 0
1 0 1 0 1
0 0 0 0 1
1 1 1 1 0
Enter the heuristic values of the nodes
1.2
2.3

3.1

4.8
5.2

Enter the source
4
The graph is explored as follows
4   5   3   1   2
```

Note:

1. For file operations - upload files using upload button, Files will be upload to /uploads folder. You can read those files in program from /uploads folder. To write a file from your program, write files to '/myfiles' folder. Please note the uploaded files stored in the server only for the current session.

2. For detailed documentation check - Our Documentation, or check our Youtube channel.

Thanks for using our

Online Java Compiler IDE

to execute your program

GET IT ON Google Play

Download on the App Store

Know Your JDoodle

- JDoodle Supports 76+ Languages with Multiple Versions and 2 DBs. **Click here** to see all.
- Fullscreen - side-by-side code and output is available. click the "⛶" icon near execute button to switch.
- Dark Theme available. Click on "•••" icon near execute button and select dark theme.
- You can embed code from JDoodle directly into your website/blog. **Click here** to know more.
- JDoodle offers an API service. You can execute programs just by calling our API. **Click here** to know more.
- If you like JDoodle, Please share us in Social Media. **Click here** to share.
- Check our **Documentation Page** for more info.

JDoodle For Your Organisation

- Do you have any specific compiler requirements?
- Do you want to integrate compilers with your website, webapp, mobile app, courses?
- Do you need more than our **Embed** and **API** features?
- Looking for Multiple Files, Connecting to DB, Debugging, etc.?
- Are you building any innovative solution for your students or recruitment?
- Want to run JDoodle in-house?
- Custom Domain, White labelled pages for your institute?

Contact us - We are happy to help!

JDoodle is serving the programming community since 2013