

Project 1: Shortest Path

(Uninformed + Informed Search)

Our Problem: We are given a file which contains vertices and edges. We need to read those files and add them to our graph. After reading all the data in a particular file, we apply our Search Algorithms and print desired outputs and performance measures.

I choose **Dijkstra** algorithm to do uninformed search and **Astar** algorithm to do informed search.

How to run script:

```
String pathname1 = arr[i] + "/e.txt";
String pathname2 = arr[i] + "/v.txt";

try {
    File filename = new File(pathname1);
    InputStreamReader reader = new InputStreamReader(new FileInputStream(filename));
    BufferedReader br = new BufferedReader(reader);

    String line = "";
    line = br.readLine();
    while (line != null) {
        line = br.readLine();
        if(line == null) continue;
        String[] s = line.split(",");
        int a = Integer.parseInt(s[0]);
        int b = Integer.parseInt(s[1]);
        int c = Integer.parseInt(s[2]);
        if(!map.containsKey(a)) map.put(a, new LinkedList<>());
        map.get(a).add(new int[]{b, c});
        if(!map.containsKey(b)) map.put(b, new LinkedList<>());
        map.get(b).add(new int[]{a, c});
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

I have submitted the source code, I wrote comment in every place which I thought is necessary.

Result comparison : Both of them can get the optimal result, So I use these two results to check whether I wrote the right code. I recorded all steps when search the goal point, But I think the path is same so I do not want to print it.

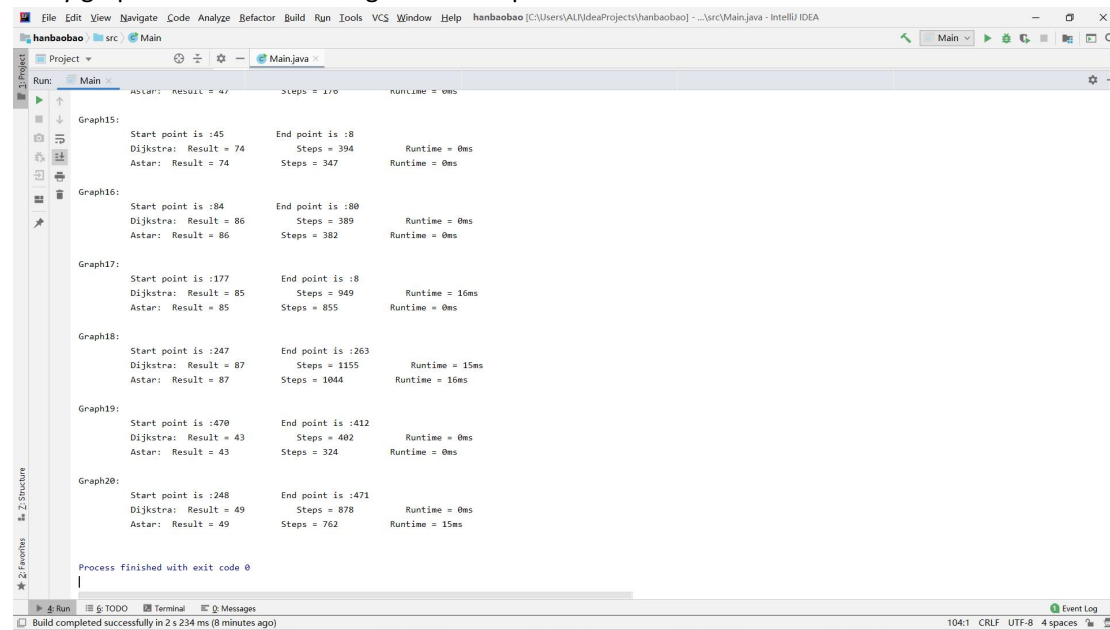
Time complexity comparison : In small graph, sometimes Dijkstra is faster than Astar. But when the graph becomes more and more complex or the two points are far away, Astar is faster than Dijkstra. And I believe that when the graph becomes very large, Astar is the best choice.

These two pictures are the result of graph2000_0.4 :

Graph18:

Start point is :247	End point is :263	
Dijkstra: Result = 87	Steps = 1155	Runtime = 15ms
Astar: Result = 87	Steps = 1044	Runtime = 16ms

Every graphs' result and running time in this picture:



Performance Measure: Dijkstras take $O(V^2)$ time in the worst case, but we can't calculate A* time complexity as it depends on the heuristic we choose. We shall compare them by the practical time taken and number of steps taken. Here are few screen shots where we can see A* takes less number of steps then Dijkstra.