

Implementation Ford Bellman - Java

Java - AdvDataStructure/Algorithms/com/ShortPath/FB.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- AdvDataStructure
 - JRE System Library [JavaSE-9]
 - Algorithms
 - AVL
 - BST
 - com.GraphAlgorithm
 - BFSGraph.java
 - DFSGraph.java
 - Graph.java
 - NumberReading.java
 - print.java
 - TopologicalSort.java
 - TopologicalSortGraph.java
 - com.GraphAlgorithm.TopologicalSort
 - com.ShortPath
 - BellmanFord.java
 - DijkstraAlgorithmSet.java
 - DijkstraQueue.java
 - FB.java
 - FloydWarshall.java
 - Greedy
 - HashMap
 - MergeSort
 - Other
 - PQueue
 - RedBlack

FB.java

```
1 package com.ShortPath;
2 import java.util.*;
3
4 public class FB {
5     static final int INF = Integer.MAX_VALUE / 2;
6
7     public static class Edge {
8         int v, weight;
9
10        public Edge(int v, int cost) {
11            this.v = v;
12            this.weight = cost;
13        }
14    }
15
16    public static boolean bellmanFord(List<Edge>[] graph, int s, int[] dist, int[] pred) {
17        Arrays.fill(pred, -1);
18        Arrays.fill(dist, INF);
19        dist[s] = 0;
20        int n = graph.length;
21        boolean updated = false;
22        for (int step = 0; step < n; step++) {
23            updated = false;
24            for (int u = 0; u < n; u++) {
25                if (dist[u] == INF) continue;
26                for (Edge e : graph[u]) {
27                    if (dist[e.v] > dist[u] + e.weight) {
28                        dist[e.v] = dist[u] + e.weight;
29                        dist[e.v] = Math.max(dist[e.v], -INF);
30                        pred[e.v] = u;
31                        updated = true;
32                    }
33                }
34            }
35            if (!updated)
36                break;
37        }
38        // if updated is true then a negative cycle exists
39    }
40 }
```

Crete Edge to contain vertex and weight

Steps:

- 1- start with weight graph
- 2- choose start vertex and assign infinity path value to all other vertices
- 3- visit all edge and apple to relax the path distances

Formula FB: / Relaxation Formula :

```
relax (Edge u v) {
    if (distance[v] > distance[u] + weight[u,v])
        (distance[v] > distance[u] + weight[u,v])
        parent [v] = u
}
```

4- after all vertices have path lenghts , you will check if negative cycle or not

Console

<terminated> FB [Java Application] C:\Program Files\Java\jdk-9\bin\javaw.exe (Jan 27, 2018, 6:24:41 PM)

1 2 31

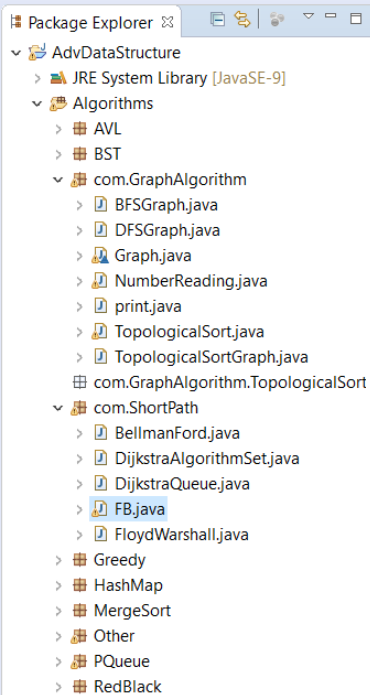
Writable Smart Insert 1 : 24

Task List

Find All Activ...

Outline

- com.ShortPath
 - FB
 - INF : int
 - Edge
 - v : int
 - weight : int
 - bellmanFord(List<Edge>[] graph, int s, int[] dist, int[] pred)
 - findNegativeCycle
 - main(String[] args)



```

41
42 public static int[] findNegativeCycle(List<Edge>[] graph) {
43     int n = graph.length;
44     int[] pred = new int[n];
45     Arrays.fill(pred, -1);
46     int[] dist = new int[n];
47     int last = -1;
48     for (int step = 0; step < n; step++) {
49         last = -1;
50         for (int u = 0; u < n; u++) {
51             if (dist[u] == INF) continue;
52             for (Edge e : graph[u]) {
53                 if (dist[e.v] > dist[u] + e.weight) {
54                     dist[e.v] = Math.max(dist[u] + e.weight, -INF);
55                     dist[e.v] = Math.max(dist[e.v], -INF);
56                     pred[e.v] = u;
57                     last = e.v;
58                 }
59             }
60             if (last == -1)
61                 return null;
62         }
63         for (int i = 0; i < n; i++) {
64             last = pred[last];
65         }
66         int[] p = new int[n];
67         int cnt = 0;
68         for (int u = last; u != last || cnt == 0; u = pred[u]) {
69             p[cnt++] = u;
70         }
71         int[] cycle = new int[cnt];
72         for (int i = 0; i < cycle.length; i++) {
73             cycle[i] = p[--cnt];
74         }
75         return cycle;
76     }
77 }
78 // ----- Main -----

```

If you found Negative Cycle Then Time $O(VE)$

Negative weight edges can create negative weight cycles - cycle which will reduce the total path distance by coming back to the same point

Steps

- 1- start filling distance array with infinity
- 2- filling predecessor array with 0
- 3- start apply relax formula for all edges by n-1

Formula FB / Relax Formula:

```

relax (Edge u v){
    if (distance[v] > distance[u] + weight[u,v])
        (distance[v] > distance[u] + weight[u,v])
        parent [v] = u
        last = v
}

```

- 4- check if graph has any negative cycle

Console

<terminated> FB [Java Application] C:\Program Files\Java\jdk-9\bin\javaw.exe (Jan 27, 2018, 6:24:41 PM)

1 2 3



- Package Explorer
- AdvDataStructure
 - JRE System Library [JavaSE-9]
 - Algorithms
 - AVL
 - BST
 - com.GraphAlgorithm
 - BFSGraph.java
 - DFSGraph.java
 - Graph.java
 - NumberReading.java
 - print.java
 - TopologicalSort.java
 - TopologicalSortGraph.java
 - com.GraphAlgorithm.TopologicalSort
 - com.ShortPath
 - BellmanFord.java
 - DijkstraAlgorithmSet.java
 - DijkstraQueue.java
 - FB.java
 - FloydWarshall.java
 - Greedy
 - HashMap
 - MergeSort
 - Other
 - PQueue
 - RedBlack

```
64     last = pred[last];
65 }
66 int[] p = new int[n];
67 int cnt = 0;
68 for (int u = last; u != last || cnt == 0; u = pred[u]) {
69     p[cnt++] = u;
70 }
71 int[] cycle = new int[cnt];
72 for (int i = 0; i < cycle.length; i++) {
73     cycle[i] = p[--cnt];
74 }
75 return cycle;
76 }
77 // ----- Main -----
78
79
80 public static void main(String[] args) {
81     List<Edge>[] graph = new List[4];
82     for (int i = 0; i < graph.length; i++) {
83         graph[i] = new ArrayList<>();
84     }
85     graph[0].add(new Edge(1, 1));
86     graph[1].add(new Edge(0, 1));
87     graph[1].add(new Edge(2, 1));
88     graph[2].add(new Edge(3, -10));
89     graph[3].add(new Edge(1, 1));
90     int[] cycle = findNegativeCycle(graph);
91     System.out.println(Arrays.toString(cycle));
92 }
93 }
```

You will determine Edges and
send value for vertex and weight

Console

```
<terminated> FB [Java Application] C:\Program Files\Java\jdk-9\bin\javaw.exe (Jan 27, 2018, 6:24:41 PM)
[1, 2, 3]
```

Quick Access

Task List

Find

All Activ...

Outline

com.ShortPath

FB

INF : int

Edge

v : int

weight : int

Edge(int, int)

bellmanFord(List<

findNegativeCycle

main(String[]) : vo

Writable

Smart Insert

40 : 3