模版·Bellman-Ford(求单源最短路径，权值可负但不可环)

```
class Edge{
    int v;
    int weight;

    Edge(int v,int weight){
        this.v=v;
        this.weight=weight;
    }
}
class Graph{
    int V;
    LinkedList<Edge>[] adjList;

    Graph(int V){
        this.V=V;
        adjList=new LinkedList[V];
        for(int i=0;i<V;i++){
            adjList[i]=new LinkedList<>();
        }
    }

    void addEdge(int u,int v,int weight){
        adjList[u].add(new Edge(v,weight));
    }

    static void bellmanFord(Graph graph,int src){
        int V=graph.V;
        int[] dist=new int[V];

        Arrays.fill(dist,Integer.MAX_VALUE);
```

```
        dist[src]=0;

        for(int i=1;i<V;i++){
            for(int u=0;u<V;u++){
                for(Edge edge:graph.adjList[u]){
                    int v=edge.v;
                    int weight=edge.weight;
                    if(dist[u]!=Integer.MAX_VALUE&&dist[u]+weight<dist[v]){
                        dist[v]=dist[u]+weight;
                    }
                }
            }
        }

        for(int u=0;u<V;u++){
            for(Edge edge:graph.adjList[u]){
                int v=edge.v;
                int weight=edge.weight;
                if(dist[u]!=Integer.MAX_VALUE&&dist[u]+weight<dist[v]){
                    System.out.println("图中包含负权重环路");
                    return;
                }
            }
        }

        printSolution(dist,src);
    }

    static void printSolution(int[] dist,int src){
        System.out.println("从源点"+src+"到各顶点的最短距离:");
        for(int i=0;i<dist.length;i++){
            if(dist[i]==Integer.MAX_VALUE){
                System.out.println("到顶点"+i+"的距离:INF");
```

```
                }else{
                    System.out.println("到顶点"+i+"的距离:"+dist[i]);
                }
            }
        }
}
public class BellmanFord{
    public static void main(String[] args){
        int V=5;
        Graph graph=new Graph(V);

        graph.addEdge(0,1,-1);
        graph.addEdge(0,2,4);
        graph.addEdge(1,2,3);
        graph.addEdge(1,3,2);
        graph.addEdge(1,4,2);
        graph.addEdge(3,2,5);
        graph.addEdge(3,1,1);
        graph.addEdge(4,3,-3);

        Graph.bellmanFord(graph,0);
    }
}
```