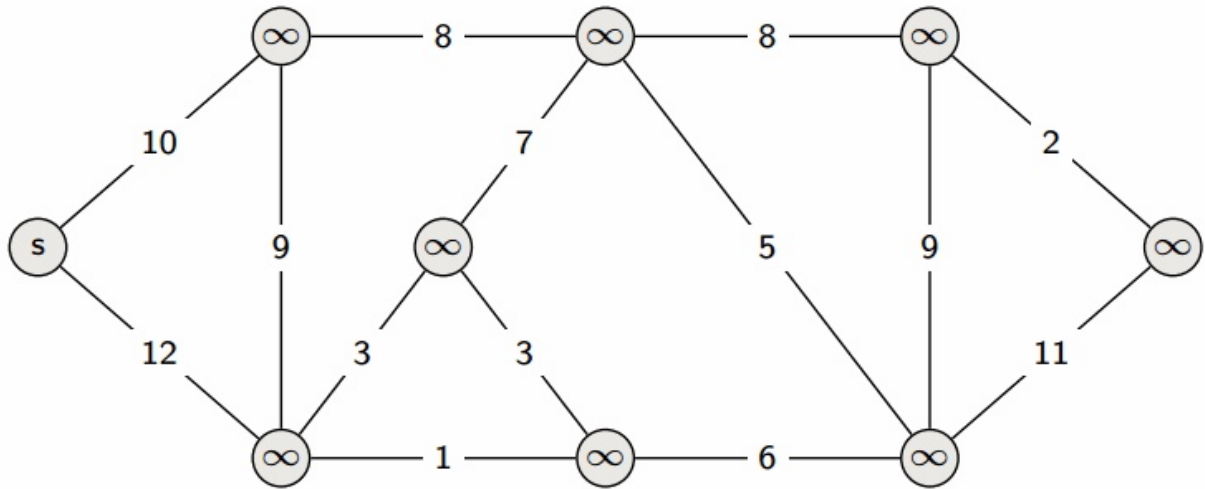


## 다익스트라(Dijkstra) 알고리즘



다익스트라(Dijkstra) 알고리즘은 단일 출발점 최단 경로 알고리즘 중 하나로, 음의 가중치가 없는 그래프에서 출발 노드에서 다른 모든 노드까지의 최단 경로를 구하는 알고리즘입니다. 이 알고리즘은 "탐욕적인 기법(greedy technique)"을 사용하여 최단 거리를 구합니다.

동작 방식은 다음과 같습니다.

1. 출발 노드를 기준으로 최단 거리를 저장하는 배열을 선언합니다. 출발 노드는 0으로 초기화하고, 다른 노드는 무한대로 초기화합니다.
2. 아직 처리하지 않은 노드 중에서 출발 노드에서 가장 가까운 노드를 선택합니다.
3. 해당 노드를 거쳐서 다른 노드로 가는 거리가 기존보다 더 짧으면 최단 거리를 갱신합니다.
4. 모든 노드를 방문할 때까지 2~3 과정을 반복합니다.

이를 우선순위 큐(priority queue)를 사용하여 구현하면 더욱 빠른 속도로 실행할 수 있습니다.

**Dijkstra ( with Java )**

```

import java.util.*;

class Dijkstra {
    static class Edge implements Comparable<Edge> {
        int vertex;
        int weight;

        public Edge(int vertex, int weight) {
            this.vertex = vertex;
            this.weight = weight;
        }

        @Override
        public int compareTo(Edge o) {
            return weight - o.weight;
        }
    }

    static int INF = Integer.MAX_VALUE;
    static List<Edge>[] graph;
    static int[] dist;

    public static void dijkstra(int start) {
        PriorityQueue<Edge> pq = new PriorityQueue<>();
        dist[start] = 0;
        pq.offer(new Edge(start, 0));

        while(!pq.isEmpty()) {
            Edge edge = pq.poll();
            int vertex = edge.vertex;
            int weight = edge.weight;

            if(dist[vertex] < weight) continue;

            for(Edge nextEdge : graph[vertex]) {
                int nextVertex = nextEdge.vertex;
                int nextWeight = nextEdge.weight + weight;

                if(nextWeight < dist[nextVertex]) {
                    dist[nextVertex] = nextWeight;
                    pq.offer(new Edge(nextVertex, nextWeight));
                }
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int V = sc.nextInt();
        int E = sc.nextInt();
        int start = sc.nextInt();

        graph = new ArrayList[V+1];
        for(int i=1; i<=V; i++) {
            graph[i] = new ArrayList<>();
        }

        dist = new int[V+1];
        Arrays.fill(dist, INF);
    }
}

```

위 코드는 우선순위 큐를 이용한 Dijkstra 알고리즘 구현 코드입니다. 입력 값으로는 정점의 수( $V$ ), 간선의 수( $E$ ), 시작 정점 ( $start$ )이 주어지며, 간선 정보는  $u, v, w$ 의 형태로 입력됩니다. 출력 값으로는 시작 정점에서 각 정점까지의 최단 거리를 출력합니다.