https://jeonyeohun.tistory.com/96

기본 개념 ->

어떤 vertex x부터 어떤 vertex y까지의 최단 거리가 d(x,y)라 하면

최단 거리를 구성하는 path 사이에 지나가는 vertex를 r이라고 한다.

그러면 모든 r에 대해서 d(x,y) = d(x,r) + d(r,y)가 성립하게 된다.

즉 shortest path는 해당 경로에 속하는 subpath들 모두 shortest path로 구성되어야 한다.

이것이 거짓이라면 x->r, r->y까지의 sub shortest path가 d(x,y)에 포함되지 않아야 하며, 이는 포함되지 않는 sub shortest path를 이용해 다시 path를 구성한다면 본래의 d(x,y)보다 짧은 path가 만들어지게 된다. 이는 본래의 가정인 optimal solution이 d(x,y)라는 가정에 거짓임으로 shortest path는 해당 path을 구성하는 vertex끼리의 sub shortest path로 구성되어 있다는 것을 알 수 있다.

즉 문제를 해결하기 위해선 vertex x로부터 vertex y까지의 거리가 m일때 이 m보다 작은 거리값을 반환하는 vertex u가 발견된다면 d(x,u)+d(u,y)로 거리값을 업데이트 해야한다.

다익스트라 알고리즘 – dp 활용, weight값이 음수인 경우 활용할 수 없다.

https://m.blog.naver.com/ndb796/221234424646

shortest path는 path 사이의 vertex 끼리의 sub shortest path들로 이루어져있다는 점을 활용한다. 탐색을 진행하면서 현재 저장된 path 값보다 더 작은 값으로 구성될 수 있는 path값을 발견하면 업데이트한다.

일단 vertex 1개를 고정시킨뒤 그 vertex와 연결된 간선을 탐색하여 해당 값을 배열에 저장(연결되지 않은 vertex와의 거리는 무한대)