

최단거리 Dijkstra $O(m \log n)$

= Prim 이랑 똑같은데 간선을 선택할 때 시작점에서부터의 가중치가 가장 작은 것을 선택

R : 정답인 노드 집합

B : R 에 있는 노드만 거쳐서 갈 수 있는 노드 집합

D_{min} : 시작노드부터의 최단거리

처음에 시작점을 R 에 추가.

B 에서 D_{min} 이 가장 작은 노드를 R 로 변경

그렇게 새로 들어온 노드 u 와 인접한 노드들 w 에 대해 D_{min} 갱신.

$D_{min}(w)$ 와 $D_{min}(u) + u$ 에서 w 까지의 거리 둘 중 작은 값을 D_{min} 으로 갱신.

왜냐하면, 그 이전에 R 에 있던 노드를 통해 갈 수 있는 최단거리는 이미 갱신되어 D_{min} 에 반영되어 있기 때문이다.

증명.

Base : 노드가 1개일 때는 자명하게 성립.

Step: R 에 V_0, \dots, V_i , B 에 w_0, \dots, w_j 인 상황이다.

여기서 B 중, D_{min} 이 가장 작은 노드를 R 로 바꾼다.

이것이 추가될 노드가 아니라면, 현재 R 만 거쳐서 오는 것이 최단 경로가 아니고, B 를 더 거쳐서 오는 더 짧은 길이 있다는 것이다.

그럼 1 경로는 처음으로 B 에 넘어가는 것이 있을 텐데, 그 길이는 알고리즘이 추가한 노드로 가는 것보다 항상 기쁘고, 모순이다.

(D_{min} 이 가장 작은 노드를 선택했기 때문이다)

우선순위 큐로 구현.

우선순위 큐에 다 넣어버리고, 꺼낼때 이미 R이 속하면 버리고, B 중에 제일 작은 값을 선택할 수 있다.