

< GBC_Algorithm PA 7 >

27기 최 하영

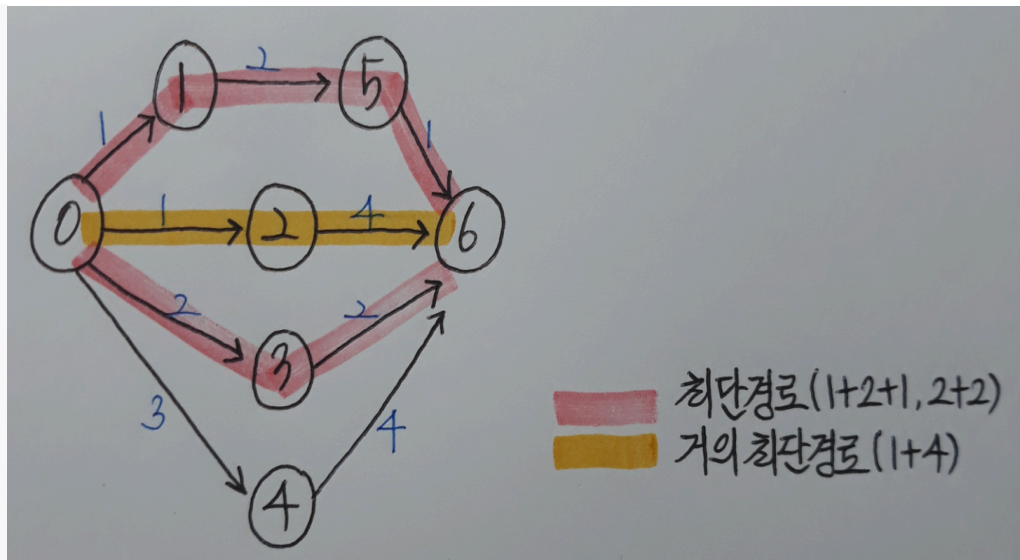
백준 5719 _ 거의 최단 경로

1. Fact

- 거의 최단 경로
→ 최단 경로에 포함되지 않는 도로로만 이루어진 경로 중 가장 짧은 것
- 한 집에는 공유기 1개만 설치 가능, 가장 인접한 두 공유기 사이의 거리 최대
- 입력 : 장소의 수 N , 도로의 수 M
시작점 S 도착점 D
도로의 정보 U, V, P (U 에서 V 로 가는 도로의 길이는 P)
- 출력 : 거의 최단 경로의 길이 (없을 경우, -1 출력)

2. Overviews

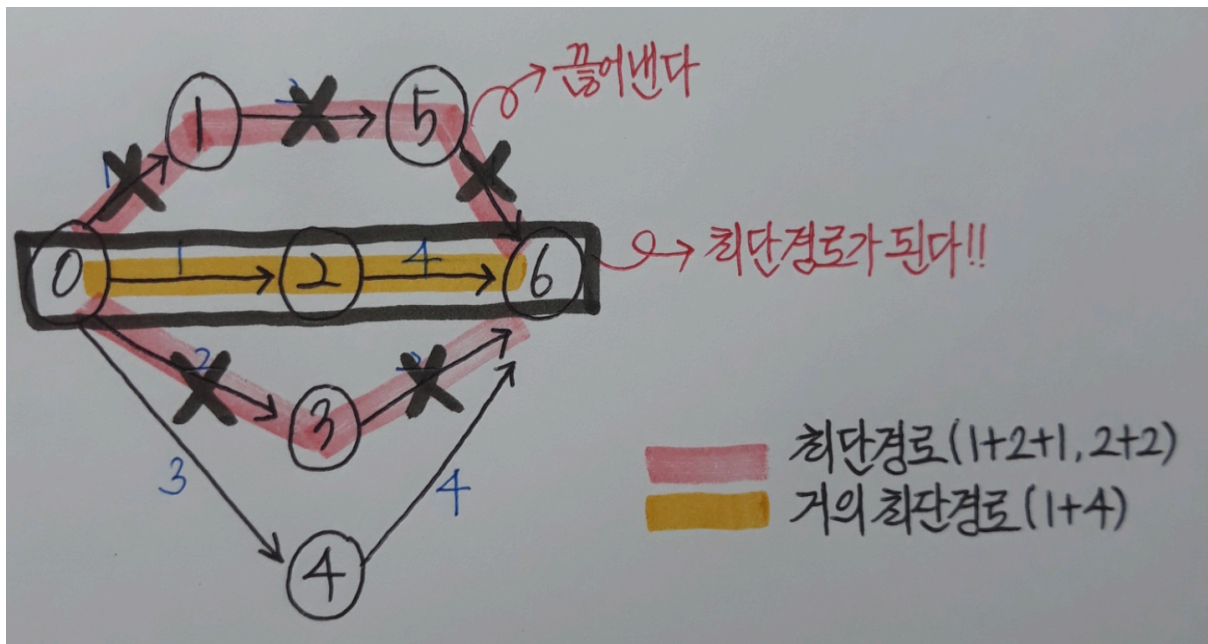
```
7 9
0 6
0 1 1
0 2 1
0 3 2
0 4 3
1 5 2
2 6 4
3 6 2
4 6 4
5 6 1
```



주어진 TEST CASE 1번을 그려보면 위와 같다. (node 7, edge 9)

최단 경로는 $0 \rightarrow 1 \rightarrow 5$ (4), $0 \rightarrow 3 \rightarrow 6$ (4)이고,

거의 최단 경로는 $0 \rightarrow 2 \rightarrow 6$ (5)이다.



가중치가 있는 그래프에서 최단거리를 찾아서 해당 노드를 연결하는 엣지를 끊고!
 다시 최단거리를 찾아서 해당 가중치들을 저장한 값을 리턴 해주면!
 거의 최단 경로의 값을 찾아주지 않을까?!

생김새를 보니까 BFS로 탐색하는 것이 좋아 보인다
 어떻게 구현하지...

3. Algorithm

- 1) TEST CASE와 같이 값을 입력 받는다.
- 2) 다익스트라를 이용해서 최단 경로를 찾는다.
- 3) 최단거리인 node들을 보관해준다.
- 3) BFS를 이용해 최단 경로를 -1로 초기화하면 삭제하는 효과
- 4) 다익스트라를 이용해서 다시 최단 경로 찾는다.
- 5) 해당 최단 경로는 처음 최단 경로를 제거한 후이므로 거의 최단 경로가 된다
 코드로 구현하면 헛갈린다...

4. Time complexity

$O(N^2) \rightarrow$ 인접 행렬로 표현한 그래프! (N은 정점의 수)