

Algorithm

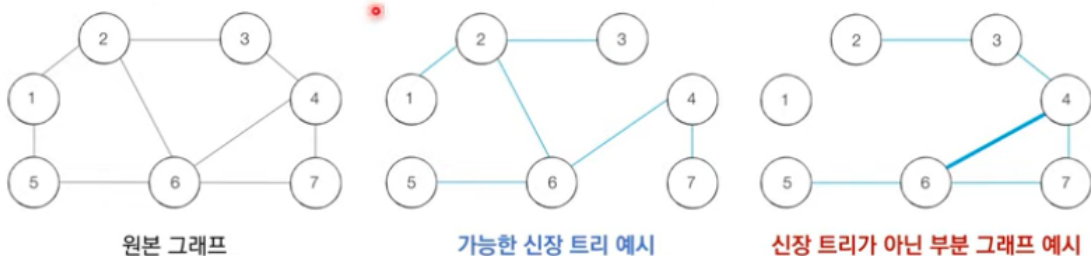
[Algorithm] 35강 : 크루스칼 알고리즘의 정의와 구현

2020. 11. 26. 00:08 수정 삭제 공개

크루스칼 알고리즘

1.1 신장 트리란?

- 그래프에서 모든 노드를 포함하면서 사이클이 존재하지 않는 부분 그래프를 의미한다.
- 모든 노드가 포함되어 서로 연결되면서 사이클이 존재하지 않는다는 조건은 트리의 조건이기도 하다.



1.2 최소 신장 트리

- N개의 도시가 존재하는 상황에서 두 도시 사이에 도로를 놓아 전체 도시가 서로 연결될 수 있게 도로를 설치하는 경우를 예시로 든다.

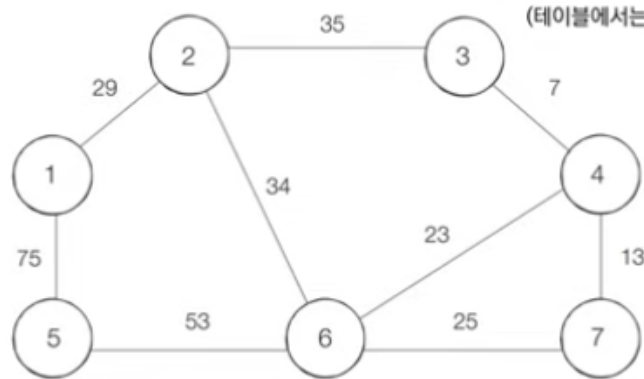


1.3 크루스칼 알고리즘이란?

- 대표적인 최소 신장 트리 알고리즘이다.
- 그리디 알고리즘으로 분류된다.
- 구체적인 동작은 다음과 같다
 - 간선 데이터를 비용에 따라 오름차순으로 정렬
 - 간선을 하나씩 확인하며 현재의 간선이 사이클을 발생시키는지 확인
 - 사이클이 발생하지 않는 경우 최소 신장 트리에 포함시킨다.
 - 사이클이 발생하는 경우 최소 신장 트리에 포함시키지 않는다.
 - 모든 간선에 대하여 2번의 과정을 반복한다.

1.4 동작 과정 살펴보기

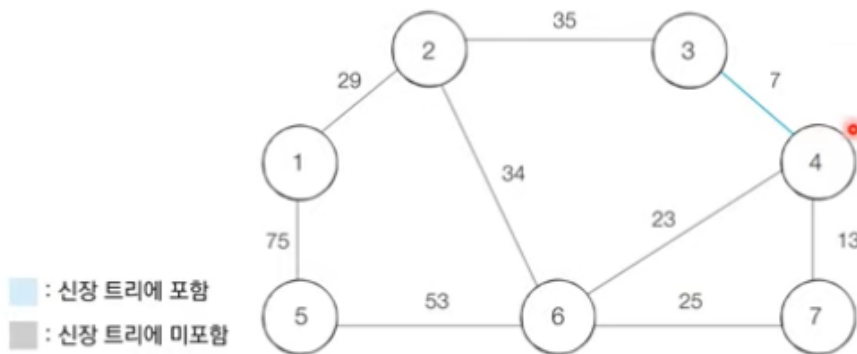
- [초기 단계] 그래프의 모든 간선 정보에 대하여 오름차순 정렬을 수행합니다.



(테이블에서는 가독성을 위해 간선에 따라서 나열했습니다.)

간선	(1, 2)	(1, 5)	(2, 3)	(2, 6)	(3, 4)	(4, 6)	(4, 7)	(5, 6)	(6, 7)
비용	29	75	35	34	7	23	13	53	25

- [Step 1] 아직 처리하지 않은 간선 중에서 가장 짧은 간선인 (3, 4)를 선택하여 처리합니다.

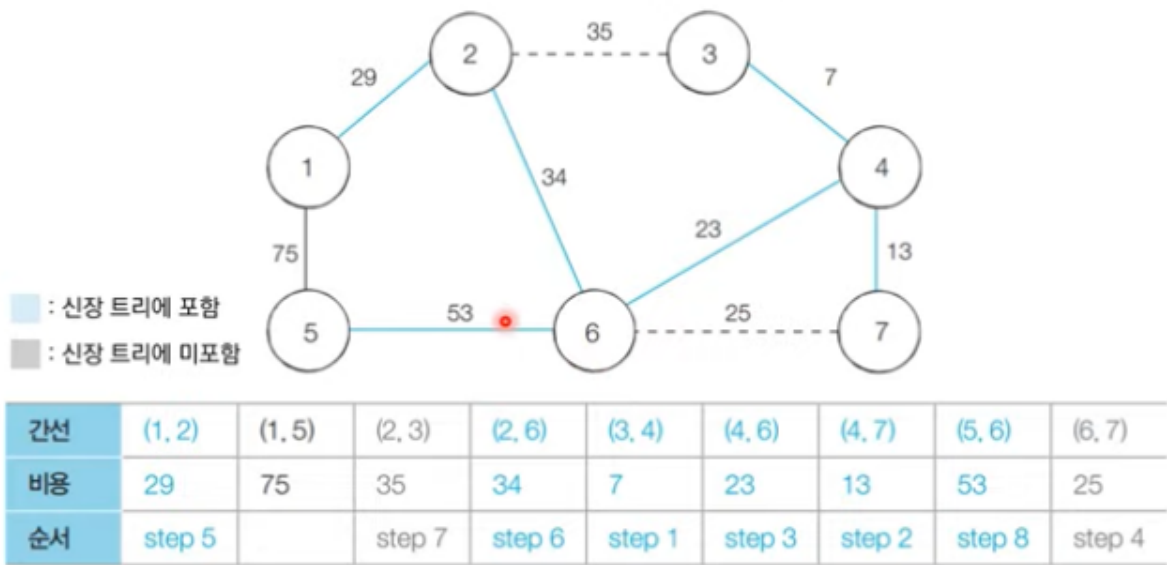


■ : 신장 트리에 포함
■ : 신장 트리에 미포함

간선	(1, 2)	(1, 5)	(2, 3)	(2, 6)	(3, 4)	(4, 6)	(4, 7)	(5, 6)	(6, 7)
비용	29	75	35	34	7	23	13	53	25
순서					step 1				

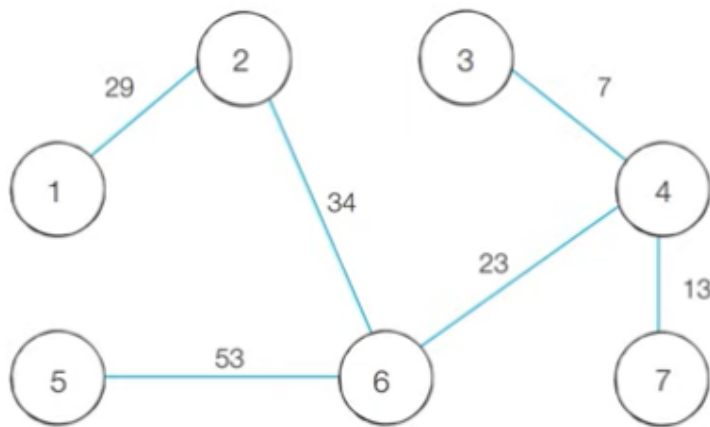
이 과정을 반복한다.

- [Step 8] 아직 처리하지 않은 간선 중에서 가장 짧은 간선인 (5, 6)을 선택하여 처리합니다.



이것의 결과로

- [알고리즘 수행 결과]
 - 최소 신장 트리에 포함되어 있는 간선의 비용만 모두 더하면, 그 값이 최종 비용에 해당합니다.



이 신장트리를 만들게 된다.

1.5 구현

```
# 특정 원소가 속한 집합을 찾기
def find_parent(parent, x):
    # 루트 노드를 찾을 때까지 재귀 호출
    if parent[x] != x:
        parent[x] = find_parent(parent, parent[x])
```

```

return parent[x]

# 두 원소가 속한 집합을 합치기
def union_parent(parent, a, b):
    a = find_parent(parent, a)
    b = find_parent(parent, b)
    if a < b:
        parent[b] = a
    else:
        parent[a] = b

# 노드의 개수와 간선의 개수 입력받기
v, e = map(int, input().split())
parent = [0] * (v + 1) # 부모 테이블 초기화 하기

edges = []
result = 0

# 부모 테이블상에서, 부모를 자기 자신으로 초기화
for i in range(1, v + 1):
    parent[i] = i

for _ in range(e):
    a, b, cost = map(int, input().split())
    # 비용순으로 정렬하기 위해서 튜플의 첫 번째 원소를 비용으로 설정
    edges.append((cost, a, b))

# 간선을 비용순으로 정렬
edges.sort()

for edge in edges:
    cost, a, b = edge
    # 사이클이 발생하지 않는 경우에만 집합에 포함
    if find_parent(parent, a) != find_parent(parent, b):
        union_parent(parent, a, b)
        result += cost
print(result)

```

이 자료는 동빈 나 님의 **이코 테** 유튜브 영상을 보고 정리한 자료입니다.

[참고 : www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC](http://www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC)

[Algorithm] 35강 : 크루스칼 알고리즘의 정의와 구현□

[Algorithm] 34강 : 서로소 집합을 활용한 사이클 판별□

[Algorithm] 33강 : 서로소 집합 자료구조의 정의와 구현□

[Algorithm] 32강 : 최단 경로 알고리즘 기초 문제 풀이□

[Algorithm] 31강 : 플로이드 워셜 알고리즘의 정의와 구현□

[Algorithm] 30강 : 다익스트라 최단 경로 알고리즘의 정의와 구현□

크루스칼 알고리즘

크루스칼 알고리즘 구현

크루스칼 알고리즘 정의



나아무늘보

혼자 끄적끄적하는 블로그 입니다.