

예시) $N=5$ $M=3$

arr 인접리스트 =

0	2	1,3	2	5	4
---	---	-----	---	---	---

 visit 방문리스트 =

0	0	0	0	0	0
---	---	---	---	---	---

```
for i in range(1,5)
    if visit[i]==0 (방문한적X)
        if arr[i]이 0이면, 혼자 조를 꾸린다
            & 방문표시
```

```
for j in arr[i]:
    bfs 탐색(무리찾기 & 방문표시)
```

[SWEA] 5249 최소 신장트리

예시) $V=2$ $E=3$ arr =

(1,1),(2,1)	(0,1),(2,6)	(0,1),(1,6)
-------------	-------------	-------------

프림

가중치 weight =

0x ffff	0x ffff	0x ffff
------------	------------	------------

부모노드 parent =

0	1	2
---	---	---

방문 visit =

0	0	0
---	---	---

 (확정여부)

i) 임의의 시작점 ex) 1

weight[1] = 0

q = [(0,1)]

ii) Q가 비어질 때까지 반복

w, v = heappop(q)

[v 방문체크 (방문했으면 안된다)
 visit[v] = 1

이때 해당 정점의
 가중치가 확정된다
 ↳ 카운트!

for node, value in arr[v]

if not visit[node] and weight[node] > value:

key[node] = value

parent[node] = v

heappush(q, (value, node))

크루스칼 부모노드 parent =

0	1	2
---	---	---

arr =

(0,1,1)
(0,2,1)
(1,2,6)

i) 가중치 기준
 오름차순 정렬

i) 정점의 루트를 찾는 함수 find_set(v)

while parent[v] != v 내 부모가 나야 root이다

v = parent[v]

return v

ii) arr을 순회한다. 단, 사이클이 생기지 않게 간선의 수는 $V-1$ 개이다 (종료조건)

u, v, w = arr

u와 v의 루트를 찾고 루트가 다르면, parent[u의 root] = v의 root

↳ 간선 확정 (간선 개수 + 1)
 가중치 w 확정 (+가중치)

[SWEA] 5250 최소비용

예시) N = 3 arr =

0	2	1
0	1	1
1	1	1

가중치 weight =

0	0x ffff	0x ffff
0x ffff	0x ffff	0x ffff
0x ffff	0x ffff	0x ffff

방문 visit =

0	0	0
0	0	0
0	0	0

(확정여부)

i) 출발점 (0,0)에서 delta 이용하여 인접점점확인

• 인접점점의 높이와 출발점높이 비교!

인접점점 > 출발점 : 가중치는 높이차이+1

인접점점 < 출발점 : 가중치는 1

ii) 계산된 가중치와 기존의 가중치 비교

& 확정안된점점인지 확인

→ 기존의 가중치가 더 크면 갱신!

& 우선순위 큐에 넣기

가중치가 작은 순서로 POP 하기위해

[SWEA] 14193 통신병민코씨

예시) X = 6 arr =

#	#	#	#	#	_
#	A	#	A	#	#
#	_	#	_	A	#
#	S	_	_	#	#
#	#	#	#	#	#

Y = 5

- 1) 모든 정점이 인접점점이다.
- 2) 모든 정점이 연결된 최소신장트리 길이를 구해야 한다. ∴ 가중치 & 트리배열 생성
- ↳ 시작점은 S노드

아주큰값 0 으로 default 세팅
- i) 해당 노드로 부터 다른 A노드의 좌표와 거리구하기 (BFS)

(확정하지 않은) (가중치)
- ii) 그 거리가 이미 기록된 거리보다 짧다면, 갱신 (값 수정)

(우선순위 큐에 추가)
- 추후, 우선순위 큐에서 POP될 때 그 정점·가중치가 확정된다

• 트리배열에 1표시

• 누적가중치 계산 (거리)