5주차 요약 정리

**Minimal Spanning Tree (최소 신장 트리)**

**정의**

가중치가 있는 연결 그래프에서, 모든 정점을 포함하면서 사이클이 없고,  
간선 가중치의 총합이 최소인 트리.

**조건**

1. 모든 정점을 포함해야 함
2. 사이클이 없어야 함
3. 가중치의 합이 최소여야 함
4. 간선 수 = 정점 수 – 1

MST 알고리즘 3가지텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

시간 복잡도텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**Quiz**

1. 그래프에 같은 가중치를 가진 간선이 있다면, 최소 신장 트리는 하나만 존재한다.

-> False

-> 동일 가중치 간선으로 인해 MST는 여러 개일 수 있습니다.

2. 다음 중 MST(최소 신장 트리) 알고리즘에서 중복된 간선 가중치를 처리하는 올바르지 않은 방법은 무엇인가요?

a. 각 간선에 아주 작은 값을 더해서 가중치를 구분한다

✅ 가능. tie-break 용으로 쓰일 수 있음

b. 간선 선택 순서를 미리 정의한 리스트로 결정한다

✅ 가능. 일관된 tie-breaking에 사용

c. 중복 가중치를 가진 모든 간선을 제외한다

❌ **틀림.** 이렇게 하면 중요한 간선이 빠질 수 있어 MST를 만들 수 없게 됨

3. MST의 비용(총 가중치)은 사용된 알고리즘에 관계없이 항상 동일합니다.

-> True

4.

(1) 모든 구성 요소에서 동시에 가장 저렴한 외곽선을 선택합니다. → Borůvka 알고리즘

(2) 단일 정점에서 시작하여 가장 저렴한 외곽선을 트리에 반복적으로 추가합니다. → Prim 알고리즘

(3) 가중치가 증가하는 순서대로 외곽선을 선택하고 순환을 피하면서 추가합니다. → Kruskal 알고리즘

5. 다음 중 최소 신장 트리(MST)에 관해 사실이 **아닌 것은 무엇입니까?**

하나를 선택하세요:

a. 사이클이 없습니다.

b. 그래프의 모든 정점을 포함합니다.

c. 모서리의 총 무게가 최소화됩니다.

d. 가장 큰 가중치를 가진 에지로만 구성됩니다. ❌

6.

(1) 다음 MST 알고리즘 중 어느 것이 병렬로 (여러 구성 요소에서 동시에) 에지를 선택합니까? → Borůvka 알고리즘

(2) 어느 알고리즘이 하나 이상의 특정 정점에서 시작하지 않습니까? → Kruskal 알고리즘

7. 모든 모서리 가중치가 다른 그래프가 주어졌을 때, 시작 정점을 다르게 하여 Prim 알고리즘을 여러 번 실행하면 MST도 달라집니다. -> 거짓

﻿