**Quiz**

**📌 2주차: 그래프 기초 용어**

* **동형(Isomorphic)**: 레이블만 바꿔서 동일하게 만들 수 있는 그래프
* **동일(Equal)**: 정점 집합과 인접 관계가 동일한 그래프
* **경로(Path)**: 정점과 간선이 번갈아 등장하며, 정점이 중복되지 않음
* **차수(Degree)**: 정점에 인접한 정점 수
* **걷기(Walk)**: 정점과 간선이 번갈아 나오는 시퀀스 (중복 가능)
* **연결 그래프**: 모든 정점 쌍 사이에 경로가 존재하는 그래프
* **부분 그래프**: 특정 정점과 간선만을 포함한 그래프
* **무향 그래프**: 인접성이 상호적인 그래프
* **슐리히트 그래프(Schlicht)**: 루프/중복 간선 없음
* **다중 그래프(Multi)**: 두 정점 사이에 여러 간선 가능

**📌 3주차: 트리와 코드**

* **비순환 그래프 = Forest**
* **프루퍼 코드 길이 5 → 정점 수 = 7**
* **트리의 조건 중 동등하지 않은 것**: e. 최소한 하나의 사이클 포함
* **프루퍼 코드 생성 시 제거되는 정점**: 가장 작은 라벨의 잎
* **인접 행렬**: 0과 1로 인접 표현
* **인접 리스트**: 정점 ↔ 간선 관계 표현
* **Forest(트리 3개, 정점 9개)** → 간선 = 6
* **MST 알고리즘 비교**:
  + Boruvka: 구성 요소마다 가장 저렴한 간선
  + Prim: 정점 하나에서 시작
  + Kruskal: 가중치 오름차순, 사이클 피함

**📌 4주차: BFS**

* ❌ **모든 그래프에 신장 트리 존재** → False
* ✅ **연결 그래프에만 신장 트리 존재** → True
* **BFS 라운드**: 전선 확장
* **BFS 출력**: 트리(Tree)
* **시작점에 따라 BFS 트리는 달라짐** → True
* **미사용 간선 남음** → 회로 존재
* **부분 그래프 + 모든 정점 포함** → 신장 그래프
* **BFS 용어**:
  + expanded vertex: 방문 완료 & 이웃 탐색 완료
  + front vertex: 방문 O, 이웃 미탐색
* **시간 복잡도**: O(|V| + |E|)

**📌 5주차: MST**

* ❌ **간선 가중치 같으면 유일한 MST X** → False
* ❌ **중복된 가중치 간선 모두 제거** → 비유효
* ✅ **MST 비용은 알고리즘과 무관하게 동일** → True
* **병렬 간선 선택 알고리즘**: Boruvka
* **시작 정점 없이 작동**: Kruskal
* ❌ **Prim 시작점 바뀌면 MST 바뀜** → False
* ❌ **MST은 가장 큰 가중치 간선으로만 구성** → False

**📌 6주차: DFS와 절단점**

* **DFS 기본**: 깊이 우선 탐색 → 진전 없으면 backtrack
* **Backtracking 조건**: 이웃 정점 모두 방문 시
* **용어 매칭**:
  + Block: 절단점 기준으로 나뉜 부분 그래프
  + Bridge: 연결 끊는 모서리
  + k-connected: 독립 경로 최소 k개
  + 독립 경로: 내부 정점 공유 X
* ❌ **블록에 절단 정점 반드시 포함 X** → 유효 X
* ❌ **절단점 지나면 독립 경로 O** → False
* **Low Point**: DFS + 사용되지 않은 간선으로 도달 가능한 최소 DFS 번호
* **Cut Vertex**: 그래프 연결을 끊는 정점

2주차

1. 그래프를 비교하기 위한 정의

* 한 그래프의 레이블을 변경하여 다른 그래프와 같게 만들 수 있다면 이 두 그래프를 동형(Isomorphic)이라고 합니다.
* 두 그래프가 같은 정점 집합을 가지고 인접 관계가 집합과 같으면 두 그래프는 동일(Equal)이라고 합니다.

2. 기본 정의

* 연속적으로 인접한 항목의 정점-모서리-정점-모서리-…-정점 형태의 시퀀스로, 어떤 정점도 두 번 이상 나타나지 않는 경우를 경로(path)라고 합니다.
* 정점에 인접한 정점의 수를 차수(degree)라고 합니다.
* 연속적으로 인접한 항목의 정점-모서리-정점-모서리-…-정점 형태의 시퀀스를 걷기(walk)라고 합니다.

3. 그래프의 다양한 유형을 무엇이라고 부릅니까?

* 두 정점 사이에 경로가 존재하는 무향 그래프를 연결 그래프(connected graph)라고 합니다.
* 주어진 그래프의 특정 정점과 그 정점을 연결하는 몇 개의 간선을 포함하는 그래프를 부분 그래프(subgraph)라고 합니다.
* 모든 인접성이 상호적인 그래프를 무향 그래프(undirected graph)라고 합니다.
* 루프도 여러 개의 간선도 포함하지 않는 그래프를 슐리히트 그래프(schlicht graph)라고 합니다.
* 같은 정점 쌍 사이에 두 개 이상의 간선이 있을 수 있는 그래프를 다중 그래프(muti graph)라고 합니다.

3주차

1. 비순환 그래프는 Forest라고도 불립니다.

2. prufer code의 길이가 5이면 원래 그래프에는 정점이 몇 개 있습니까?

* + (n+2) = 7개

3. 방향이 없는 트리에 대한 주어진 진술 중 동등하지 않은 진술을 선택하세요.

a. 연결되어 있으며, 간선보다 정점이 정확히 하나 더 많습니다.

b. 연결되고 비순환적입니다.

c. 트리의 모든 두 정점 사이에는 정확히 하나의 경로가 있습니다.

d. 루트노드가 있는 트리의 방향이 없는 대응물입니다.

e. 최소한 하나의 순환을 포함합니다.

f. 최소한으로 연결되었거나 정확히 하나의 정점으로 구성됩니다.

4. prufer 코드 생성 과정에서 각 단계에서 가장 먼저 제거되는 정점은 무엇입니까?

a. 가장 큰 라벨이 있는 잎

b. 가장 작은 라벨이 있는 잎

c. 루트 노드

d. 무작위로 선택된 내부 노드

5. 다음 설명과 용어를 일치시켜 주세요.

* 0과 1만 포함하는 무향 그래프의 발생 구조는 무엇입니까?
* 인접 행렬(Adjacency matrix)
* 간선과 정점 간의 관계를 나타내는 리스트 자료 구조는 무엇입니까?
* 인접 리스트(Adjacency list)

6. forest가 트리 3개로 이루어져 있고 정점이 9개라면 간선은 몇 개일까요?

* 정점 – 트리 갯수 = 9 – 3 = 6

7. 다음 설명과 용어를 일치시켜 주세요.

* 모든 구성 요소에서 동시에 가장 저렴한 간선을 선택합니다.
  + Boruvka 알고리즘
* 단일 정점에서 시작하여 가장 저렴한 간선을 트리에 반복적으로 추가합니다.
  + Prim 알고리즘
* 가중치가 증가하는 순서대로 간선을 선택하고 순환을 피하면서 추가합니다.
  + Kruskal 알고리즘

4주차

1. 모든 그래프에는 spanning tree가 있습니다. => False

2. 연결된 모든 그래프에는 spanning tree가 있습니다. => True

3. BFS의 각 라운드에서는 무슨 일이 일어나나요?

a. 전체 그래프를 연결합니다.

b. 현재 전선을 확장합니다.

c. 가장 깊은 노드를 먼저 탐색합니다.

d. 정점 사이의 거리를 계산합니다.

4. BFS는 어떤 출력을 만들어내나요?

a. Tree

b. loop graph

c. connected graph

d. complementary(보완) graph

5. 시작점이 다르면 BFS 트리가 변경됩니다. => True

6. 용어와 정의를 일치시키세요.

* BFS를 수행한 후 사용되지 않은 간선이 남아 있는 경우 그래프는 어떤 유형의 구조를 갖게 됩니까?
  + 회로가 포함됩니다.
* 만약 부분 그래프가 원래 그래프의 모든 정점을 포함하고 있다면, 그것은 신장 그래프(spanning graph)입니다.

7. 용어와 정의를 일치시키세요.

* BFS에서 방문이 완료되었고 모든 이웃이 탐색된 정점을 무엇이라고 부릅니까?
* expanded vertex
* BFS에서 이미 방문한 정점이지만 이웃이 아직 탐색되지 않은 정점을 무엇이라고 부릅니까?
* front vertex

8. BFS의 시간 복잡도는 무엇입니까?

a. O(|V| + |E|)

b. O(log n)

c. O(n!)

d. O(n^2)

5주차

1. 그래프에 같은 가중치를 가진 간선이 포함되어 있는 경우 고유한 최소 신장 트리는 단 하나뿐입니다. => False

2. 다음 중 MST 알고리즘에서 중복된 에지 가중치를 처리하는 데 유효 하지 않은 방법은 무엇입니까?

a. 각 정점에 매우 작은 값을 추가하여 가중치를 차별화합니다.

b. 미리 정의된 목록을 사용하여 간선 선택 순서를 결정합니다.

c. 가중치가 중복되는 모든 정점를 제외합니다.

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

4. 다음 설명과 용어를 일치시켜 주세요.

* 모든 구성 요소에서 가장 저렴한 바깥쪽 가장자리를 동시에 선택합니다.
* Boruvka 알고리즘
* 단일 정점에서 시작하여 가장 저렴한 간선을 트리에 반복적으로 추가합니다.
* Prim 알고리즘
* 가중치가 증가하는 순서대로 간선을 선택하고 순환을 피하면서 간선을 추가합니다.
* Kruskal 알고리즘

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

a. 사이클이 없습니다.

b. 그래프의 모든 정점을 포함합니다.

c. 모서리의 총 무게가 최소화됩니다.

d. 가장 큰 가중치를 가진 에지로만 구성됩니다.

6. 다음 설명과 용어를 일치시켜 주세요.

* 다음 MST 알고리즘 중 어느 것이 병렬로 (여러 구성 요소에서 동시에) 간선을 선택합니까?
  + Boruvka 알고리즘
* 하나 이상의 특정 정점에서 시작 하지 않는 알고리즘은 무엇입니까?
  + Kruskal 알고리즘

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

6주차

1. 다음 중 DFS(깊이 우선 탐색)를 수행하는 올바른 방법을 설명한 것은 무엇입니까?

a. 각 정점을 방문한 후에는 항상 루트로 돌아갑니다.

b. 항상 차수가 가장 높은 정점을 먼저 방문하세요.

c. 가중치가 가장 낮은 모서리를 선택하세요.

d. 최대한 깊이 탐구하고, 더 이상 진전이 없을 때는 뒤로 물러나세요.

2. DFS 탐색 중에 백트래킹은 언제 발생합니까?

a. 현재 정점의 모든 이웃을 방문한 경우

b. 현재 정점의 숫자가 더 작을 때

c. 그래프에 사이클이 포함된 경우

d. 새로운 정점이 발견되면

e. 대기열이 비어 있을 때

3. 다음 설명과 용어를 일치시켜 주세요.

* 그래프를 절단 정점을 기준으로 분해하여 형성된 부분 그래프입니다.
  + block
* 그래프의 연결을 끊는 제거된 모서리입니다.
  + bridge
* 두 정점이 특정 최소 개수 이상의 독립 경로로 연결된 그래프입니다.
  + k-connected
* 시작/끝을 제외하고 공통된 내부 정점을 공유하지 않는 두 경로입니다.
  + 독립 경로(Independent Path)

4. 다음 중 블록에 대한 유효한 조건이 아닌 것은 무엇입니까?

a. 절단된 정점이 포함되어 있습니다.

b. 연결되어 있습니다.

c. 최소한 하나의 모서리를 포함합니다.

5. 절단된 정점을 통과하는 두 경로는 독립적인 경로로 간주됩니다. => False

6. 다음 설명과 용어를 일치시켜 주세요.

* 정점에서 도달할 수 있는 가장 작은 DFS 번호(0개 이상의 DFS 트리 간선과 0개 또는 1개의 사용되지 않는 간선을 사용)
  + Low Point
* 그래프의 연결을 끊는 정점을 제거합니다.
  + Cut Vertex