알고리즘 0318

1교시

Subgraph : 그래프의 정점들의 부분 집합과 그들을 연결하는 간선들로 구성됩니다

Induced(유도) Subgraph : 그래프에서 정점 부분 집합에 연결된 모든 간선을 포함합니다

Path(경로) : 정점을 여러 번 사용 X, 노드간에 간선 1개씩(길이 : 간선 수)

Directed path : 방향 상관없이 한번 씩만 이동

Forward-directed path : 방향에 신경써서 이동

Walk : 필요하지 않은 경로가 생김(시간이 오래 걸림)

Shortening Walks into Paths : 짧은 경로로 이동(필요하지 않은 경로 제거)

Close walk(닫힌 행보) : 시점과 종점이 같다.(시점을 여러 번 이동해도 된다.)

Circuit(회로) : 경로와 유사, 정점을 2번 이상 사용하면 안된다.(순서는 상관 X)

Cycle(순환) : 유향 그래프 방향을 지키는 닫힌 행보(반대로 가면 안된다)

Complement(반대) : 두 그래프(서로 반대)가 간선이 겹치지 않는 경우

완전형 그래프를 보고, 다른 간선으로 구분

Iff(if and only if) : 동시에, 서로 조건이 같아야 한다. (p는 q이다, q는 p이다.)

Connectedness

* Reachabiity : 갈 수 있는지?
* Connected graph : 최대 연결된 구분(component)
* 컴퓨터는 모든 가능한 간선을 탐색한다.
* Component : Sap graph

Adjacency-bsed Data Structures

* 노드 리스트(행, 열), 관계 리스트
* Adjacency relation(인접 관계)
  + (0, 4) => 0번쨰 노드랑 4번쨰 노드랑 연결
  + 탐색시간 길다 (전체적으로 다 확인해야해서)
* Adjacency list(인접 리스트)
  + (4) => 4번쨰 노드랑 연결
  + 탐색시간 짧음(해당하는 노드만 확인하면 되서)
* Adjacency matrix(인접 행렬)
  + 행과 열로 노드 연결
  + 유향 그래프는 다른 행렬의 형태

실습 때 인접 행렬 진행

Same or different?

형태는 같아도 노드가 다르면 다른 그래프이다.

그래프 데이터 구조는 다르게 나온다.

Isology graph: 노드 순서만 바꾸면 같아지는 그래프

Check => 노드 개수가 다르면 Isology graph가 아니다.

Icidence-based Data Structures

* 노드리스트(열), 간선리스트(행), 관계 리스트
* Incidence list
* Incidence matrix (undirected vs directed)
* 위에서 바라보면 선 1개이다.

Rooted Trees: Construction

* 루트 노드로 시작, 방향이 있다.
* 루트 노드 시작기준으로 모든 가능한 결과
* 끝이 없는 정점은 없다.
* Parent -> child 관계(parent노드의 간선은 1개씩, child 노드의 간선은 여러 개)
* 컴퓨터는 탐색할 때, 역순으로 찾는다.
* 두 끝의 노드를 탐색할 때, 한 끝에서 뿌리로 가고, 다른 한 끝에서 뿌리로 가고 연결한다.
* 연결된 경로는 1개씩이다.
* 일반 그래프에 유향으로 바꾸면 트리가 보인다.
* 최소 연결 그래프 => 낭비 없는 그래프
* 간선 1개 제거하면 연결이 안된다.

2교시

Charactisation of Trees

* Root Tree (무향)
* 여러 추론이 있다.

1. G is the undirected counterpart of a rooted tree
   1. Root Tree (무향)
2. There is exactly one path between any two verices in G
   1. Root Tree는 경로 1개만 존재(트리는 회로가 없다.)
3. G is connected and acylic
   1. 트리가 여러 개 연결되면 숲이다.
4. G is minimally connected or consists of exactly one vertex
   1. 최소 연결 되어 있고, 정점의 수 – 간선의 수 = 1이다.
5. G is connected and has exactly one vertex more than edges.
   1. 1개만 빼면 연결되지 않는다.

Futher Properties of Trees

포레스트는 정점의 수 – 간선의 수 = component 수

인접 행렬은 메모리 낭비일 수 있다. => tree poor, tree coding

Prufer Codes

* 트리를 코드로 만드는 방법
* 노드, 숫자 이름 필요
* 가장 낮은 잎을 찾는다.(부모 1, 이웃 1) 삭제
* 정점 2개 남을 때까지 반복
* 뿌리의 코드를 기록하면서 삭제한다.

텍스트, 친필, 폰트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.