특강 4일차 - 그래프 이론

김태희

Contents

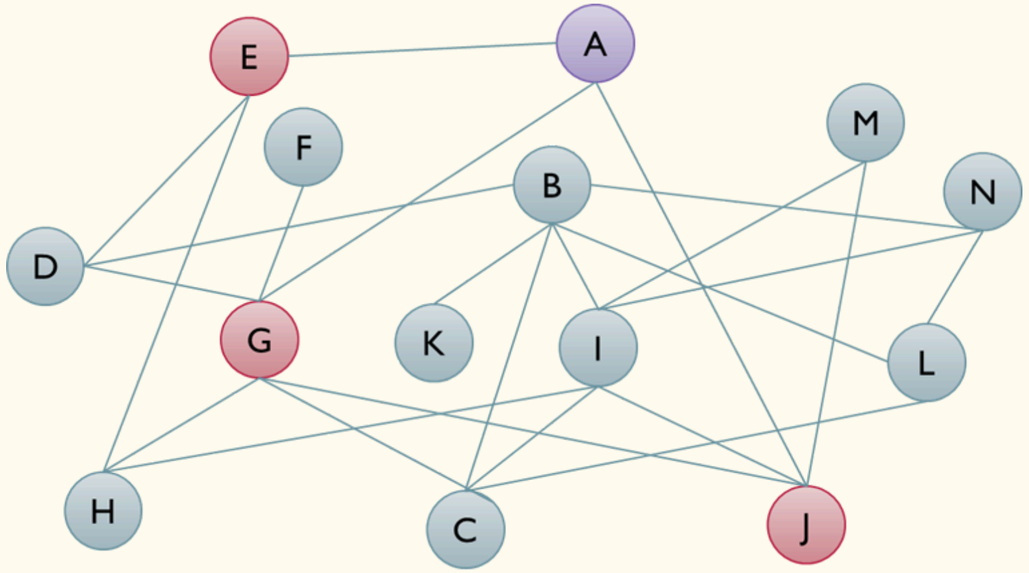
그래프

그래프

- 아이템들과 이들 사이의 연결 관계를 표현
- 정점(Vertex) : 그래프의 구성 요소로 하나의 연결점
- 간선(Edge) : 두 정점을 연결하는 선
- 차수(Degree) : 정점에 연결된 간선의 수
- 그래프는 정점(Vertex)들의 집합과 이들을 연결하는 간선(Edge)들의 집합으로 구성된 자료 구조
- M : N 관계를 가지는 원소들을 표현하기에 용이

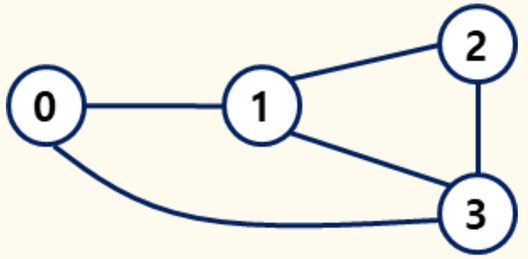
그래프

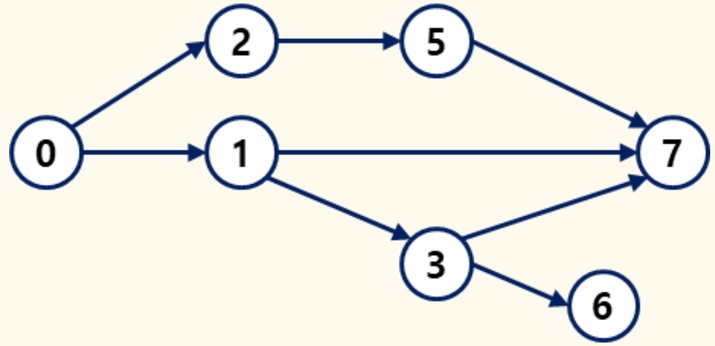
- 도시들의 정보가 다음과 같이 주어졌을 때 A와 연결된 도시 중 다른 도시와 가장 많이 연결이 되어 있는 도시는 ?
- A도시와 연결된 도시는 E, G, J임

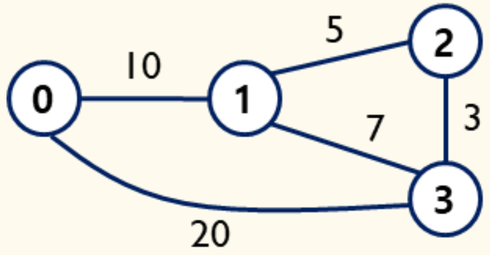


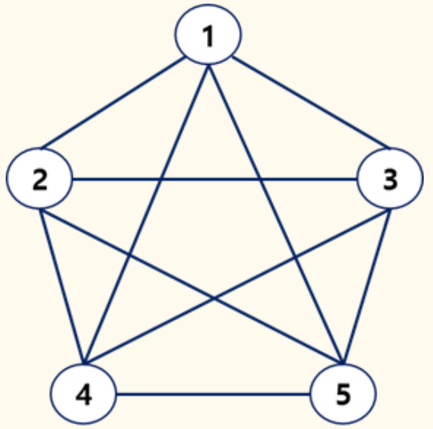
그래프 종류

- 방향성 유무에 따라
  - 유향 그래프
  - 무향 그래프
- 가중치 그래프
- 완전 그래프
  - 모든 정점이 최대의 간선을 가짐
- 부분 그래프
  - 그래프의 일부 정점이나 간선을 제거한 그래프







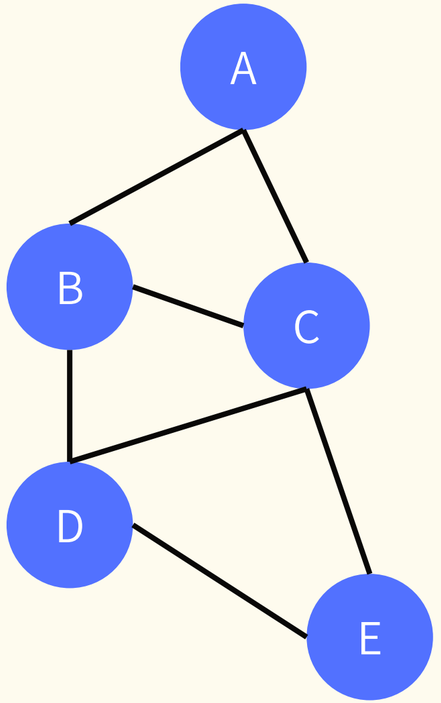


그래프의 표현

- 간선의 정보를 저장하는 방식, 메모리나 성능을 고려해서 결정
- 인접 행렬 (Adjacent matrix)
  - 정점x정점 크기의 2차원 리스트를 이용해서 간선 정보를 저장
- 인접 리스트 (Adjacent List)
  - 각 정점마다 다른 정점으로 나가는 간선의 정보를 리스트로 저장
- 간선 리스트(Edge List)
  - 간선(시작 정점, 끝 정점)의 정보를 리스트에 저장

인접 행렬

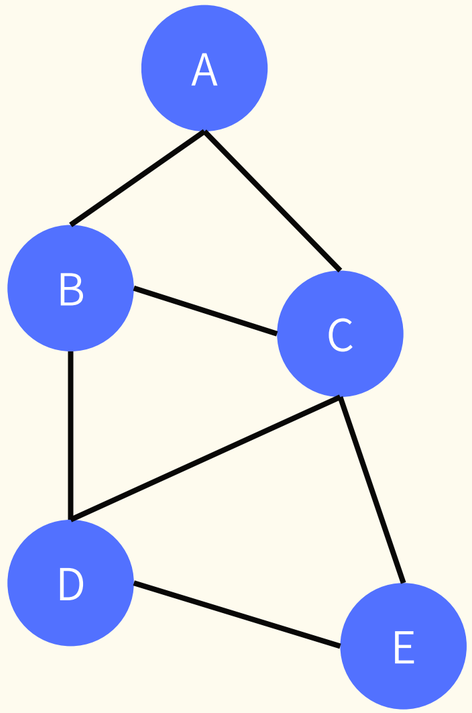
- 정점x정점 크기의 2차원 리스트를 이용해서 간선 정보를 저장
- adjMatrix[i][j] : 정점i에서 정점j로의 간선 정보 저장
  - 가중치 없는 그래프
    - 인접한 경우 : 1 또는 true
    - 인접하지 않은 경우 : 0 or false
  - 가중치 있는 그래프
    - 인접한 경우 : 가중치 값
    - 인접하지 않은 경우 : 0
- 희소 그래프의 경우 공간 효율성 나쁨

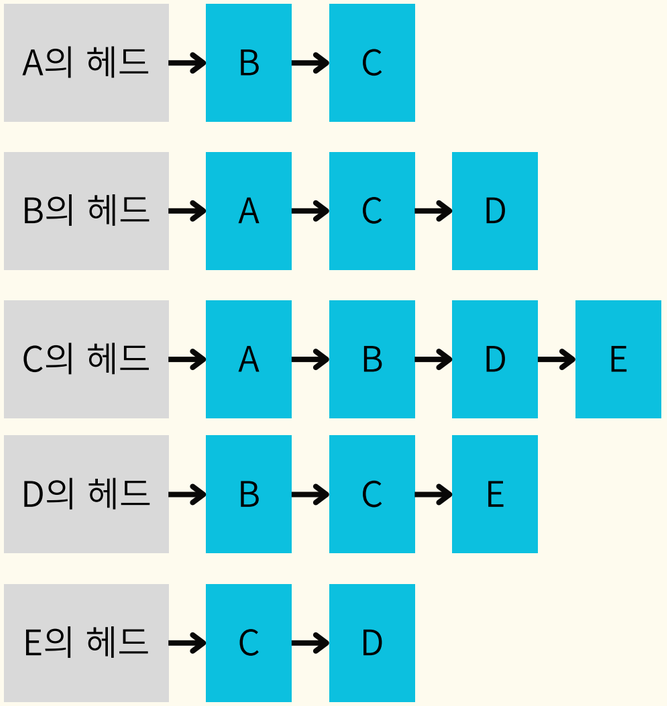


0	1	1	0	0
1	0	1	1	0
1	1	0	1	1
0	1	1	0	1
0	0	1	1	0

인접 리스트

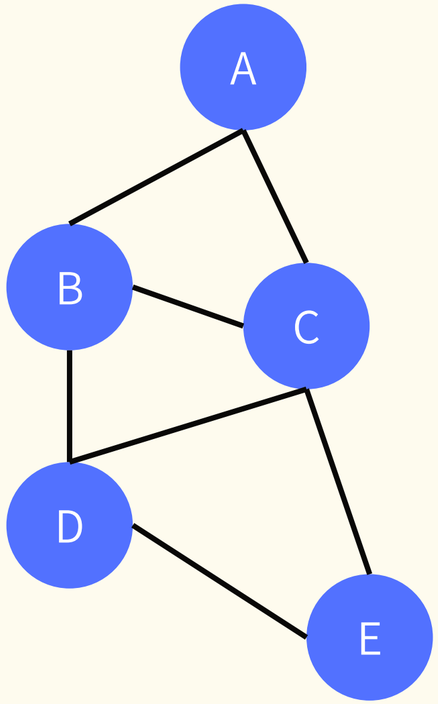
- 각 정점마다 다른 인접한 정점으로 나가는 간선의 정보를 리스트로 저장
- adjList[i] : 정점i와 인접한 정점들과의 간선 정보 리스트
- 희소 그래프의 경우 공간 효율성 좋음
- 밀집 그래프인 경우는 인접 행렬과 같은 형태가 됨





간선 리스트

- 간선(시작 정점, 끝 정점)의 정보를 리스트에 저장



A B

A C

B C

B D

C D

C E

D E

그래프 탐색

그래프 탐색

- 비선형 자료구조 탐색
- 너비 우선 탐색(Breadth First Search, BFS)
- 깊이 우선 탐색(Depth First Search, DFS)

그래프 탐색 - BFS

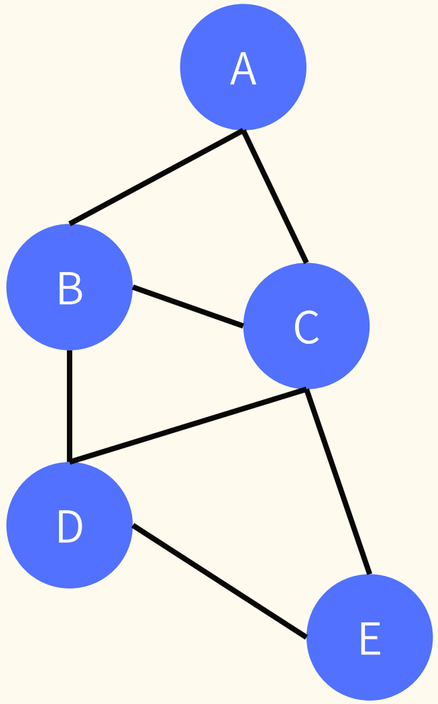
- 너비 우선 탐색(Breadth First Search, BFS)
- 탐색 시작 정점의 인접한 정점들을 먼저 모두 차례로 탐색한 후에,

  방문했던 정점들의 다시 인접한 정점들을 차례로 탐색하는 방식
- 너비 기준 선입 선출 방식을 이용하므로 큐를 활용

```
BFS(v) // 탐색 시작 정점 v
      큐 생성
      시작 정점 v를 큐에 삽입
      정점 v를 방문한 것으로 표시
      while (큐가 비어 있지 않은 경우)
            t ← 큐의 첫 번째 원소 반환
            for (t와 연결된 모든 간선에 대해)
                  u ← t의 인접 정점
                  u가 방문되지 않은 곳이면,
                  u를 큐에 넣고, 방문한 것으로 표시

end BFS()
```

그래프 탐색 - BFS

- 탐색 과정(A정점 부터 시작)



	A	B	C	D	E
visited					

Q									

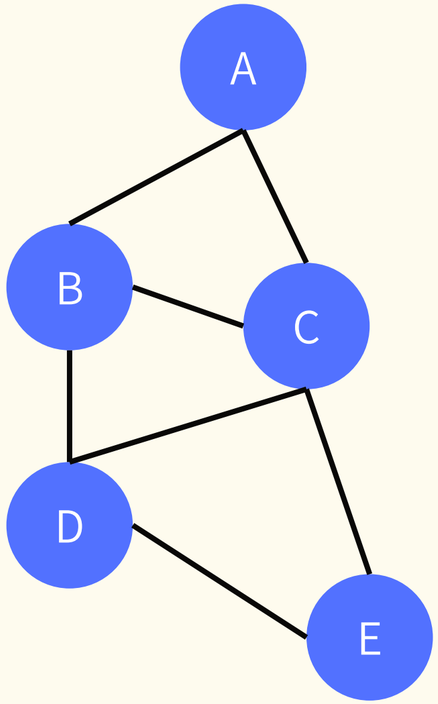
그래프 탐색 - DFS

- 깊이 우선 탐색(Depth First Search, DFS)
- 탐색 시작 정점에서 시작해 갈 수 있는 만큼 깊이 탐색한 후, 더 이상 갈 정점이 없으면 마지막에 만났던 갈림길 간선이 있는 정점으로 되돌아와서 다른 방향의 정점으로 탐색을 계속 반복하여 결국 모든 정점을 방문하는 순회 방법
- 재귀나 스택을 이용한 구현

```
DFS(v) // v:탐색 정점

    정점 v를 방문한 것으로 표시

    for (v와 연결된 모든 간선에 대해)
        u ← v의 인접 정점
        u가 방문되지 않은 곳이면,
        DFS(u)

end DFS()
```

그래프 탐색 - DFS

- 탐색 과정(A정점 부터 시작)



	A	B	C	D	E
visited					

호출 스택

최단 경로

최단 경로

- 그래프에서 탐색 가능한 경로 중 비용이 최소인 경로 찾기
- 가중치 없는 그래프
  - BFS
- 가중치 있는 그래프
  - 양의 가중치 그래프 : 다익스트라 알고리즘
  - 음의 가중치 허용 그래프 : 벨만포드 알고리즘

최단 경로

- 미로 탐색 (https://www.acmicpc.net/problem/2178)

N×M크기의 배열로 표현되는 미로가 있다.

1	0	1	1	1	1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1

미로에서 1은 이동할 수 있는 칸을 나타내고, 0은 이동할 수 없는 칸을 나타낸다. 이러한 미로가 주어졌을 때, (1, 1)에서 출발하여 (N, M)의 위치로 이동할 때 지나야 하는 최소의 칸 수를 구하는 프로그램을 작성하시오. 한 칸에서 다른 칸으로 이동할 때, 서로 인접한 칸으로만 이동할 수 있다.

위의 예에서는 15칸을 지나야 (N, M)의 위치로 이동할 수 있다. 칸을 셀 때에는 시작 위치와 도착 위치도 포함한다.

그래프 관련 실습

그래프 탐색 실습

- 바이러스
  - https://www.acmicpc.net/problem/2606
- 토마토
  - https://www.acmicpc.net/problem/7576
- 스타트 링크
  - https://www.acmicpc.net/problem/5014

Thank you!