

Chapter 6. 그래프

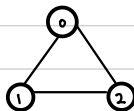
p. 320, 321

- 그래프 공집합이 아닌 정점의 집합과 간선의 집합으로 구성되는 자료구조
순환적, 사이클 허용
 $n:m$ 관계
- 무방향 그래프 (A, B)
 - 순서가 없음
 - $(A, B) = (B, A)$
 - n 개의 정점 무방향 그래프의 최대 간선 수는 $n(n-1)/2$
 - 방향 그래프 $\langle A, B \rangle$
 - A 출발 B 도착 : $\langle A, B \rangle$
B 출발 A 도착 : $\langle B, A \rangle$
 - 최대 간선 수 $n(n-1)$

그래프의 표현 1. 인접행렬 크기 n^2 , 간선 수 많은 때, 정점 많을수록 크기 커짐

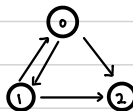
① 무방향 간선 x2, 선대칭

to \ from	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0



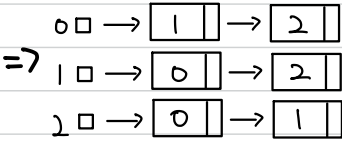
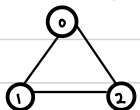
② 방향 간선 수 그대로

to \ from	0	1	2
0	0	1	1
1	1	0	1
2	0	0	0

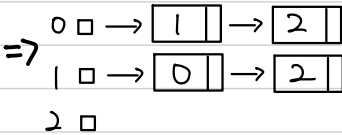
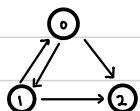


2. 인접리스트 간선 추가 적을 때 효율 (∵ 각 정점에 연결된 간선 수)

① 무방향

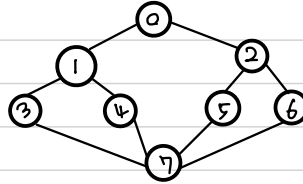


② 방향



그래프의 순회 1. 깊이 우선 순회 (스택)

- 방문하는 순서대로 정점을 스택에 쌓음, 방문이 끝나면 pop
 - 더 이상 깊이 들어가지 못할 때, 이전 정점으로 돌아와 다음 정점
- ex) 4번출발, 오름차순



노력

4 → 1 → 0 → 2 → 5 → 7 → 3 → 6

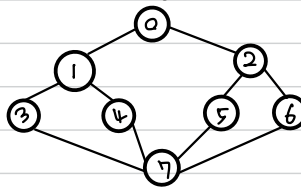
끝

7로복귀

다시 4로 복귀 ←

2. 너비 우선 순회 (큐)

- 시작 정점과 인접한 것들 우선 방문
 - 방문여정 정점을 모두 돌고 난 뒤 인접 정점 방문
- ex) 4 번 출발, 오름차순



4 → 1 → 7 → 0 → 3 → 5 → 6 → 2

방문여정

방문여정

방문여정

방문여정

다래지향적, 복귀는 없음

최소 비용 신장트리

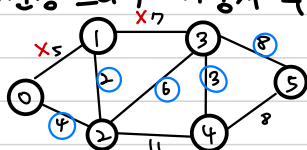
1. 모든 간선의 길이의 합이 최소인 것
2. 그래프 내의 모든 정점 포함, 간선의 수가 $n-1$ 개인 사이클이 없는 연결 그래프
3. Kruskal, Prim 알고리즘 모두 greedy method 사용

Kruskal 알고리즘

간선 위주
최소 위주
간선 수 신장합

사이클
만들어야 한다!

1. 신장 트리에 가중치 추가



정점 6개 \rightarrow 간선 5개
가중치 낮은 순서대로 선택
5, 7은 사이클이 만들어지기 때문에 X
8은 둘 다 상관없음

2	(1,2)
3	(3,4)
4	(0,2)
5	X
6	(2,3)
7	X
8	(3,5)
11	

간선 수 5개

간선 수 e 정렬에 걸리는 시간 $O(e \log_2 e)$
선택에 걸리는 시간 $O(e)$

Kruskal 시간복잡도 $O(e \log_2 e)$

\checkmark 선택 시간
무시할 것들도
+ 복잡도

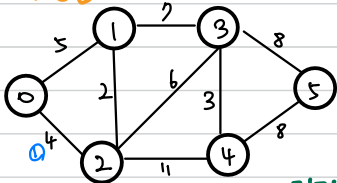
간선 수가 많을수록
 \rightarrow 더 오래 걸림

Prim 알고리즘

정점 위주
확장 위주
간선 수 신장합

1. 정점을 연결된 부분과 연결되지 않은 부분으로 나눈다.
2. 두 부분을 잇는 간선 중 최소의 간선을 선택

시작 정점 0



선택 신장트리 가중치 비교

- 4 0 5, 4 비교
- 2 0, 2 5, 2, 6, 11 비교
- 6 0, 2, 1 5, 7, 6, 11 비교 (5는 0과 연결되었기 때문에 고려 X)
- 3 0, 2, 1, 3 5, 7, 3, 8, 11 비교
- 0, 2, 1, 3, 4
- 0, 2, 1, 3, 4, 5

정점 수 n , 간선 수 $n(n-1)/2 - (n-1)$
 \Rightarrow 시간복잡도 $O(n^2)$

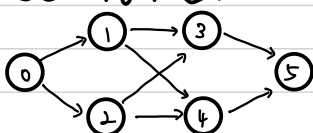
p.327, 328

작업 네트워크 방향

1. 위 상정렬

① 정점 작업 네트워크 (AOV)

- 간선 : 작업의 순서



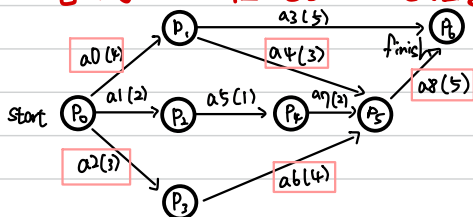
$$0 \rightarrow \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 4 \end{bmatrix} \rightarrow 5$$

둘을 합쳐줘야 다음으로 넘어간다.

② 전선 작업 네트워크 (AOE)

- 정점 : 작업의 시작과 끝 (stage, event)

- 임계경로 : 프로젝트 완료 최소시간 (최장 경로의 길이)



여유시간	작업	시간	late	early
0	a0	4	12 - (5+3+4) = 0	0
2	a1	2	12 - (5+1+4) = 2	0
0	a2	3	12 - (5+4+3) = 0	0
3	a3	5	12 - 5 = 7	4
0	a4	3	12 - (5+3) = 4	4
1	a5	1	12 - (5+2+4) = 2	2
4	a6	4	12 - (4+5) = 3	3
2	a7	2	12 - (2+5) = 5	3
0	a8	5	12 - 5 = 7	7

여유시간 = late - early

=> 여유시간 0 인 작업 (임계작업)을 모아 임계경로를 구함

제한 시간 5x