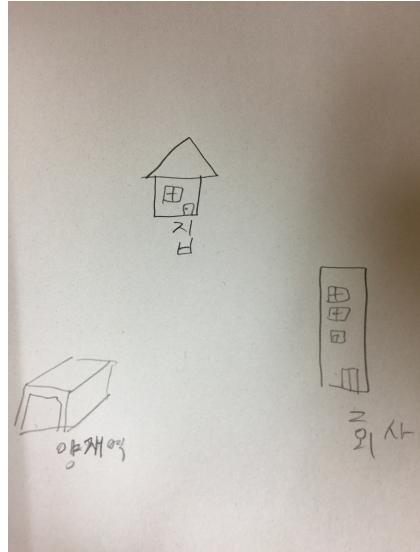
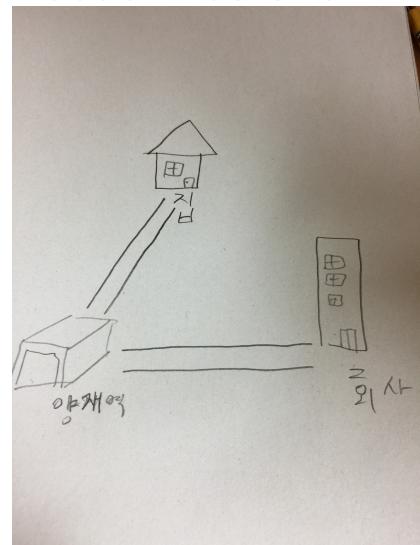


Graph

그래프에 대해서 배워봅시다. 그래프란 무엇일까요? 그래프란 Vertex(정점)과 Edge(간선)의 집합입니다. 그렇다면 Vertex 와 Edge는 무엇일까요? Vertex는 보통 State(상태), Space(장소, 혹은 공간) 등등을 나타내는데 쓰입니다. 예시를 들어 볼까요?



이렇게, 집, 양재역, 회사가 Vertex 역할을 할 수 있습니다. 이제 Edge에 대해서 배워봅시다. Edge는 두 Vertex 간의 관계를 의미합니다. 즉, 길이 될수도 있고, Vertex가 사람으로 정의한다면 친구관계나 업무관계 등으로 정의 할 수 있습니다. 보통 길이라고 생각하면 이해하기 쉽습니다. 다음과 같은 예시를 볼까요?

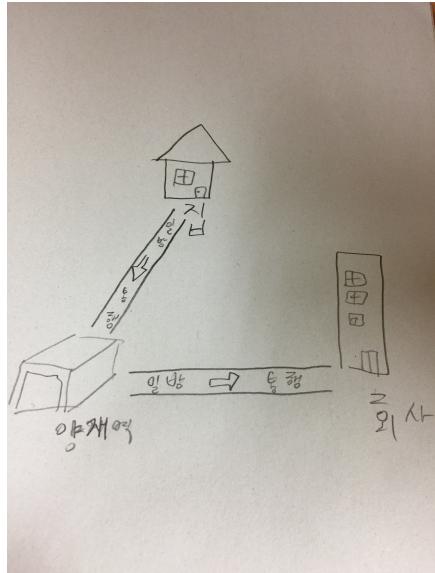


이렇게 집에서 양재역으로 향하는 길이 있다고 한다면 집과 양재역에 Edge가 있다고 할 수 있습니다. 그리고 양재역에서 회사로 향하는 길이 있다면 회사와 양재역에 Edge가 있다고 할 수 있습니다.

Vertex와 Edge의 개념에 대해서 설명했습니다. 이제 그래프의 종류중 크게 두가지로 나눌 수 있는 **directed Graph**와 **bi-directed Graph**를 알아보도록 합시다.

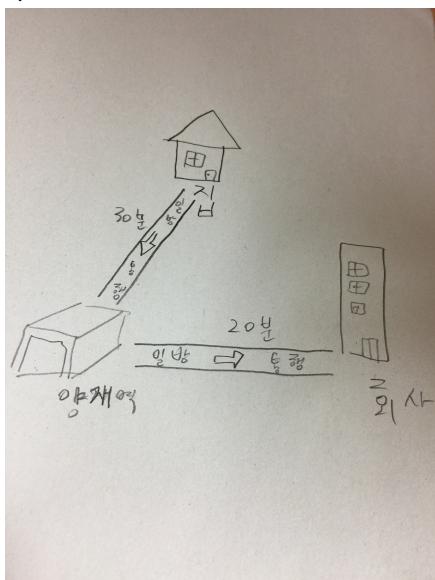
bi-directed Graph는 위 그림과 같이 방향이 없어서 양방통행이 가능함을 의미합니다. 하지만 directed Graph는 한방향으로 통행이 가능한 경우를 나타냅니다. 즉, 어떤 Edge한개를 사용하면 단방향으로 밖에 갈 수 없습니다.

다음 예시를 같이 보시죠.

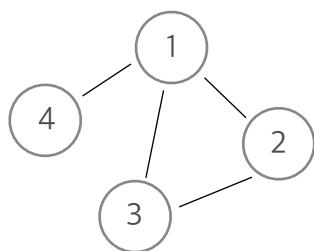


일방 통행길이 되었습니다. 이제 집에서 양재역으로 밖에 못가고 양재역에서는 회사로 밖에 못갑니다. 단방향이기 때문에 역주행은 불가능합니다. 하지만 만약에 회사에서 양재역으로 가는 일방통행길을 하나 더 만들 수 있다면 회사에서 양재역으로 갈 수 있겠지요, 물론, 그 길을 통해서 양재에서 회사로는 못갑니다.

이러한 길에 Weight(가중치)를 줄 수 있습니다. 밑에 그림처럼 말이죠. 물론 가중치 그래프는 나중에 다익스트라를 배울 때 다시 언급하도록 하겠습니다.



우리는 이제 그래프의 용어를 알아보고 그 예시를 보았습니다. 이제 활용하는 것을 봅시다. 그래프에서 Vertex는 보통 숫자로 나타냅니다. N개의 Vertex가 있다면 [1, N] 범위의 정수로 나타냅니다.



다음과 같이 말입니다. 이 경우는 bi-directed Graph 이면서 $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$ 의 Edge를 가지는 그래프가 되겠네요. 자, 이제 추가적인 용어를 몇가지 더 알아봅시다. 바로 Path(경로)와 Cycle(순환)입니다. Path란, 출발지, 도착지 $\{u, v\}$ 가 있을 때 u 에서 v 로 이동할 때 정점을 정확히 한번씩만 방문하면서 이동하는 것, 그러한 Edge의 집합을 의미합니다. 1이 출발지 3이 도착지라고 해볼까요? $1 \rightarrow 3$ 바로 가는 방법이 있고 $1 \rightarrow 2 \rightarrow 3$ 이런 식으로 가는 방법이 있습니다. 이 경우엔 $\{1, 3\}$ 과 $\{1, 3\}, \{2, 3\}$ 을 의미하게 되네요. $\{1, 4\}, \{1, 4\}, \{1, 3\}$

을 1과 3 사이의 Path라고 할 수 있을까요? 정답은 아니오입니다. 1 4 로 갔다가 4 1로 돌아온다면 1과 4를 두번 사용하게 되는 것이니까요. 그렇다면 Cycle은 무엇일까요? Cycle 은 시작지와 출발지가 같으며 그러한 Walk(이동)를 의미합니다. 예를 들어 1->4-1 도 사이클이 될 수 있으며 1->4->1->4->1 역시 사이클이 될 수 있으며 2->3->1->2 역시 사이클이 될 수 있습니다. 이때, Simple Cycle(단순 순환)이 라는것이 있습니다. 이것은 Edge와 Vertex를 정확히 한번만 사용하면서(시작점 == 출발점은 두번까지 허용) 이동하는 것을 의미합니다. 1 -> 2 -> 3 -> 1 이 Simple Cycle이라고 할 수 있겠네요. directed Graph 에서의 Path 와 Cycle의 정의는 다르지 않습니다.

여기 까지 간단한 그래프에 대한 활용이었습니다. 그렇다면 그래프의 표현은 어떻게 할까요?

컴퓨터에서의 그래프의 표현은 크게 두가지 방법이 있습니다. 바로 Adjacency Matrix(인접 행렬)와 Adjacency List(인접 리스트)입니다. Adjacency Matrix 는 두 Vertex사이에 Edge가 존재한다면 1, 존재하지 않는다면 0으로 나타냅니다. 물론 가중치가 존재한다면 1 이외의 다른 숫자로도 나타낼 수 있지만 이것에 대해서는 나중에 설명하도록 하겠습니다. 위 그래프의 Adjacency Matrix는 다음과 같이 나타낼 수 있습니다.

	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0
4	1	0	0	0

만약 directed Graph 였다면 AD[1][2] 가 1일때 AD[2][1] 이 반드시 1일 필요는 없습니다. bi-directed 였다면 반드시 1 이여야 겠지만요.

자 이제, 인접 리스트를 확인해 보아야겠죠? 인접 리스트는 어떤 Vertex 에 연결되어있는 Vertex들을 리스트로 나타내는 방법입니다. 위 그래프에서의 인접 리스트의 예시를 보여드리겠습니다.

1	2	3	4
2	1	3	
3	1	2	
4	1		

1에 연결되어있는 {2, 3, 4} 를 List[1]에 넣어주고 2에 연결되어 있는 {1, 3} 을 List[2] 에 넣어주고.. 이러한 방식입니다.

이를 통해 알 수 있는 것은 Adjacency Matrix 는 2-Dimension Array 로 구현하면 되고, Adjacency List 는 List 혹은 저번에 배운 ArrayList 등을 사용하면 됩니다! 가급적이면 구현이 간단한 ArrayList(C++에서는 vector 죠)를 사용하여 구현하는 것이 좋습니다.