

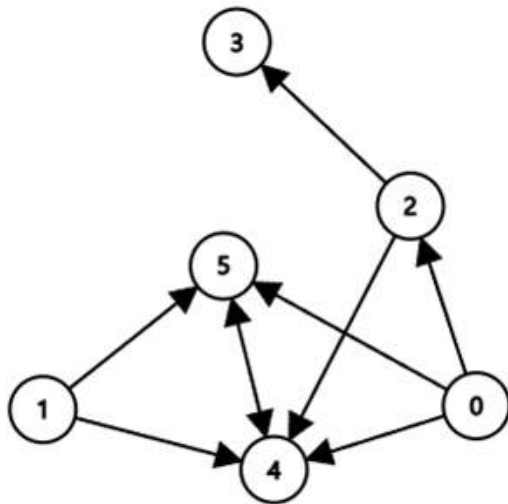
# C언어 스터디

## 6.0주차

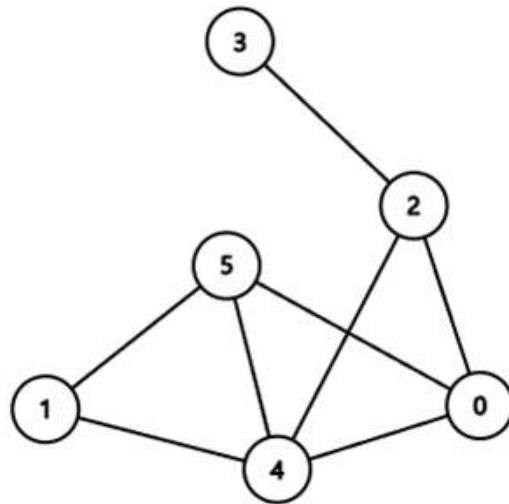
<인접리스트와 인접행렬>

CAPS

# 그래프



<방향 그래프>



<무방향 그래프>

방향그래프를 보시면 한번만에 1번에서 5번에는 갈 수 있지만 5번에서 1번을 갈 수 없습니다. 무방향그래프는 1번에서 5번, 5번에서 1번을 갈 수 있습니다. 별로 어려운 건 없습니다.

이런 그래프의 데이터를 담는 방법이 어떻게 될까요?

1. 인접 행렬
2. 인접 리스트입니다.

ind	0	1	2	3	4	5
0			1		1	1
1					1	1
2				1	1	
3						
4						1
5						

ind	0	1	2	3	4	5
0			1		1	1
1					1	1
2	1			1		
3			1		1	
4	1	1		1		1
5	1	1			1	

이게 인접행렬이 데이터를 담는 방식입니다. x에서 y로 갈 수 있다고 한다면,  $arr[x][y]=1$  이라고 처리해 주면 됩니다.

나중에 가서는 x에서 y로 가는 데 t만큼 시간이 걸린다고 한다면

$arr[x][y]=t$

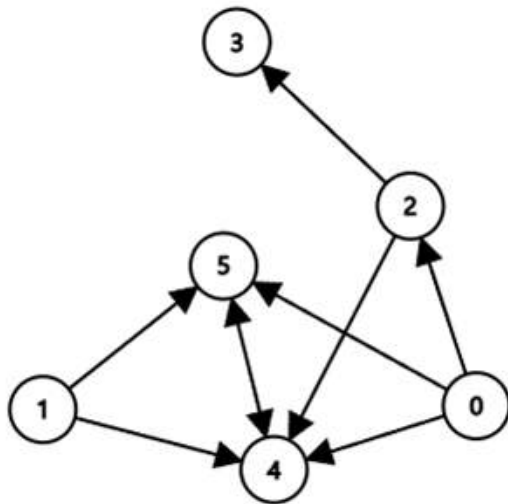
이렇게 저장해주면 됩니다.

무방향 그래프일 때는

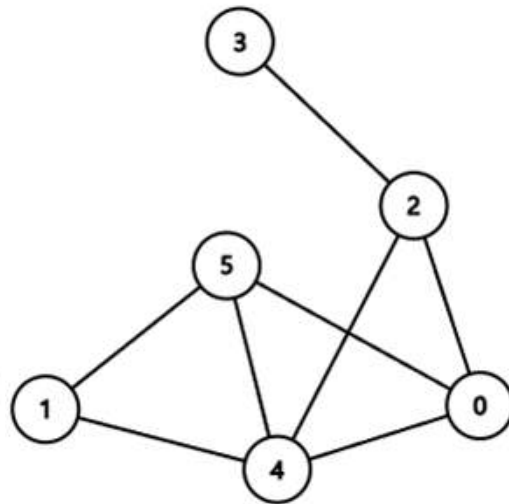
$arr[x][y]=arr[y][x]=1$

으로 저장하면 되겠죠?

# 인접리스트



<방향 그래프>



<무방향 그래프>

ind	
0	2, 4, 5
1	4, 5
2	3, 4
3	
4	5
5	

ind	
0	2, 4, 5
1	4, 5
2	3, 4
3	2, 4
4	0, 1, 3, 5
5	0, 1, 4

인접리스트는 이렇게 저장합니다.

흠 어떻게 저장해야 할까요?

```
#include<iostream>
#include<vector>
using namespace std;
vector<int> adj[6]; //adj[6] 이라는 배열의 원소를 vector로 사용합니다.
int main(void) {
    int x=0;
    int y=3;
    adj[x].push_back(y);
    adj[y].push_back(x); // 무방향 그래프라면 반대 방향도 저장해 줘야겠죠?
}
```

인접리스트에서

x에서 y로 가는 데 t만큼 걸린다고 할 때는 어떻게 저장해야할까요?

pair 자료형을 잘 쓰면 됩니다.

```
#include<iostream>
#include<vector>
#include<utility> // pair 자료형을 사용하기 위한 STL 입니다.
using namespace std;
vector<pair<int,int>> adj[6];
//adj[6] 이라는 배열의 원소를 vector로 사용합니다.
//이때 vector의 자료형을 순서쌍을 표현하는 pair 자료형을 씁니다.
//원하시면 구조체를 만들어서 쓸 수 있습니다. 하지만 나중에 pair가 편리할 때가 많습
니다.
int main(void) {
    int x=0;
    int y=3;
    int w=100;
    adj[x].push_back({y,w});
    adj[y].push_back({x,w}); // 무방향 그래프라면 반대 방향도 저장해 줘야겠죠?

    cout<<adj[x][0].first<<" "<<adj[x][0].second; // 3 100
}
```