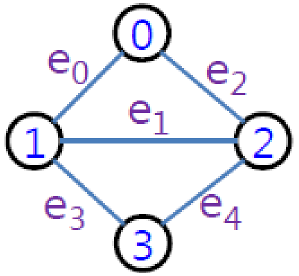
Graph Data Structure Implementation

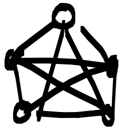
$G = (V, E)$

$|V| = n \quad |E| = m$

- Graph Data Structure
  - Adjacency Matrix



$$\underbrace{\begin{array}{c|cccc} & v0 & v1 & v2 & v3 \\ \hline v0 & 0 & 1 & 1 & 0 \\ v1 & 1 & 0 & 1 & 1 \\ v2 & 1 & 1 & 0 & 1 \\ v3 & 0 & 1 & 1 & 0 \end{array}}_{n^2 \text{ elements}}$$



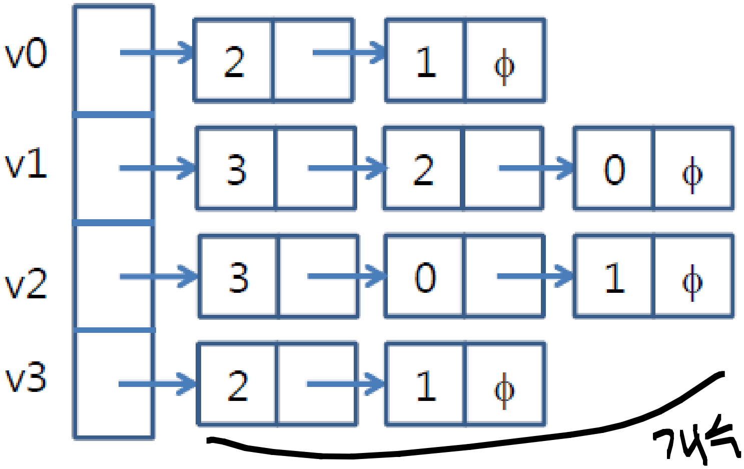
complete graph?

모든 vertex 쌍 간에 edge 가 있는 graph.

$|E| = m = {}_nC_2 = \frac{n(n-1)}{2}$

$\Rightarrow O(n^2)$

  - Adjacency List



개수 = $2m$ = $\underline{n(n-1)\text{개}}$ $\Rightarrow$ memory size $\underline{\text{data, link}}$

$2n(n-1)$

1. Graph가 sparse할 경우 adj list가 유리.
2. Vertex와 edge 정보를 저장하기 위해서는 보다 복잡한 자료구조 필요.

$\underline{m << n^2}$

① size 관점

adj matrix $O(n^2)$

adj list $\underline{O(n+m)}$

$n+2m$ elements

if $m \ll n^2$ 인 경우

sparse graph

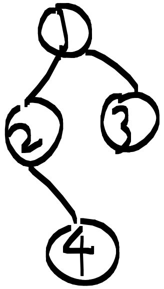
⇒ Adj List 가 유리

else $m \approx n^2$ 인 경우

Adj Matrix 가 유리.

sparse 한데도 Adj matrix 를
사용한다면?
어떤 자료구조? sparse matrix
사용.

일상생활에서 발생하는 문제의 model
대부분 sparse.



Adj Matrix

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 \end{array}$$

이부분만
기록하면 된다.

symmetric matrix

1차원 배열 사용해서

$$\underbrace{\boxed{1}\boxed{1}\boxed{0}}_{\text{row 1}}\underbrace{\boxed{0}\boxed{1}}_{\text{row 2}}\underbrace{\boxed{0}}_{\text{row 3}}$$

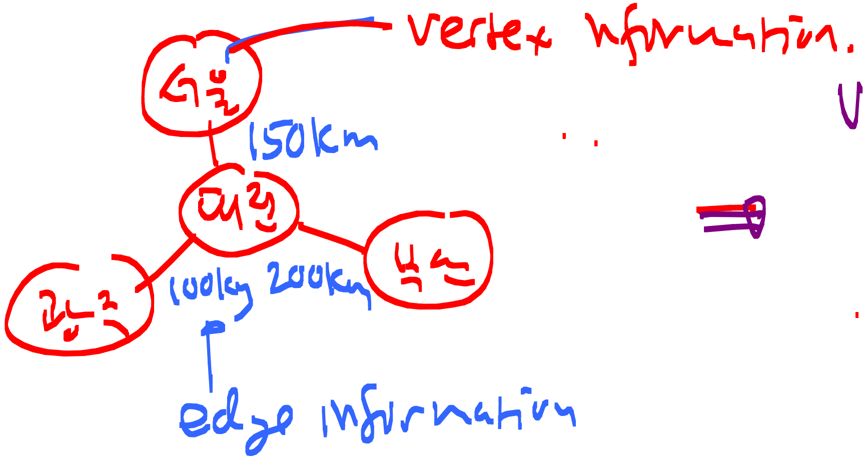
access 함수 필요

k = findindex(i, j)

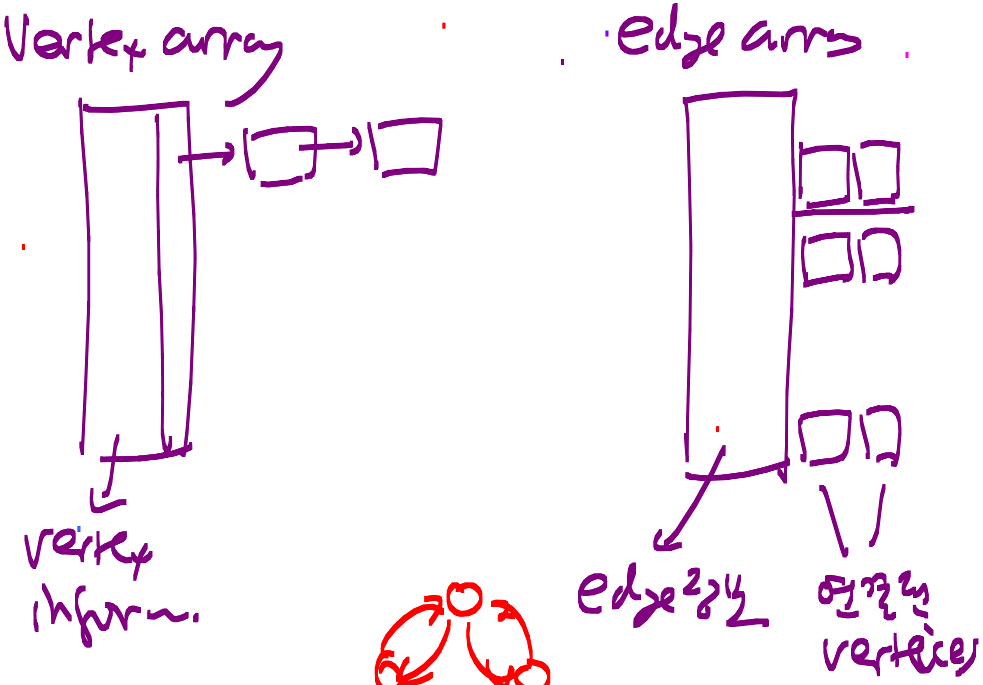
matrix index i, j 를 주면

1차원 index k를 return.

Vertex 나 edge에 정보가 있는 경우.



⇒



Directed Graph

edge에 direction이 있는 경우

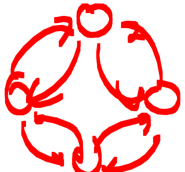
Adj matrix는 symmetric 하지 않다.

Undirected Graph

Symmetric

결론

모든 undirected graph의 집합 ⊂ 모든 directed graph의 집합



⟺