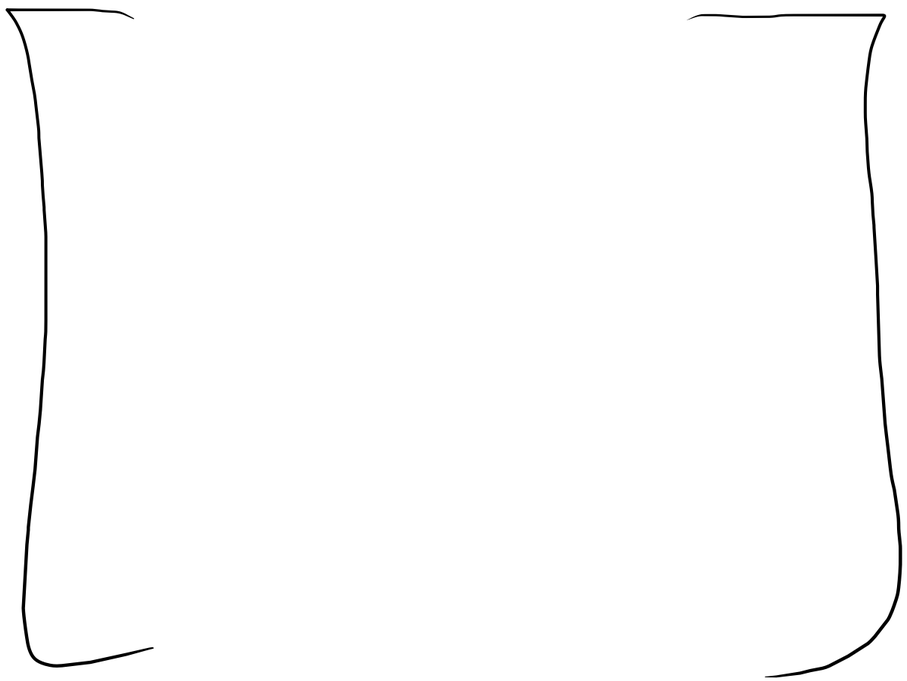
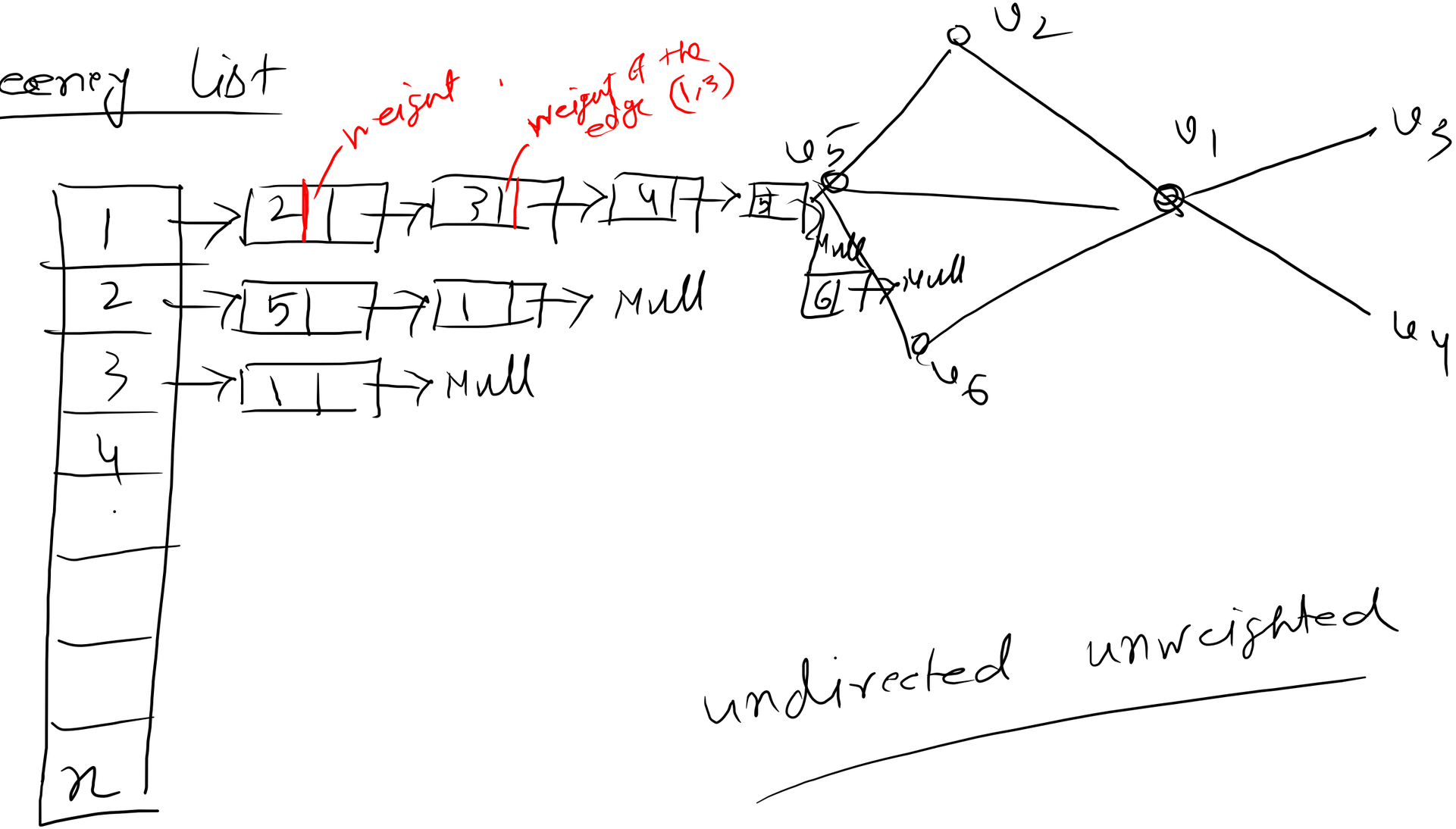
Graph representations

Adjacency matrix

$M =$ 

$n \times n$

Adjacency List



undirected unweighted

Space requirement:

- Adjacency matrix : $O(|V|^2)$ Sparse $<< |V|^2$

- Adjacency List : $O(|V| + |E|)$

Q: check whether (v_i, v_j) has an edge or not ?

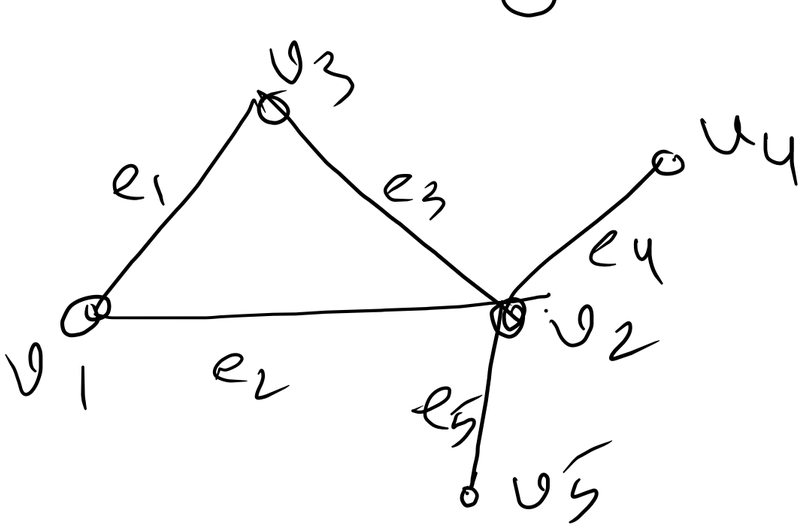
adjacency matrix - $\theta(1)$

" List $O(n)$

Incident matrix

$I = (a_{ij})_{n \times m}$

$$a_{ij} = \begin{cases} 1 & \text{if } e_j \text{ is incident on } v_i \\ 0 & \end{cases}$$



	e_1	e_2	e_3	e_4	e_5
v_1	1	1	0	0	0
v_2	0	1	1	1	1
v_3	1	0	1	0	0
v_4	0	0	0	1	0
v_5	0	0	0	0	1

$n \times m$

Graph Search / explore.

- need to visit all vertices of a graph.

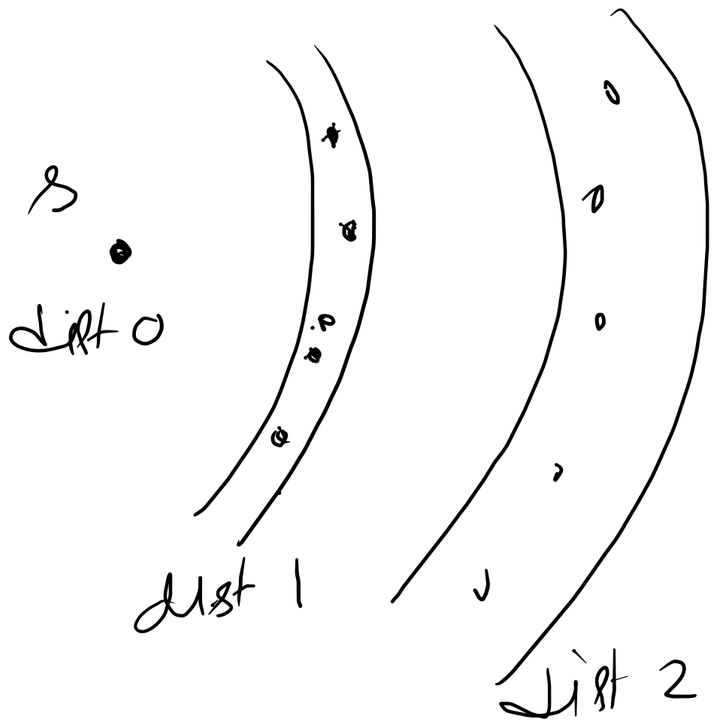
Two major searching algorithms

1) BFS - Breadth first search

2) DFS - Depth first search.

BFS

- It starts from a distinguish vertex called source.



Queue

FIFO

3 - categories of vertices.

White ← vertices are not yet explored.

'inside the queue' {
gray ← vertices are encountered but not finish its processing.

black ← vertices already explored.

BFS (G, s)

for each vertex $u \in V$

color(u) = white

$\pi[u]$ = Null

$d(u) = \infty$

color(s) = gray

$\pi[s]$ = Null

$d[s] = 0$

Enqueue (Q, s)

while ($Q \neq \phi$)

u = Dequeue (Q)

for all vertices $v \in Adj(u)$

If color(v) = white

color(v) = gray

Enqueue (Q, v)

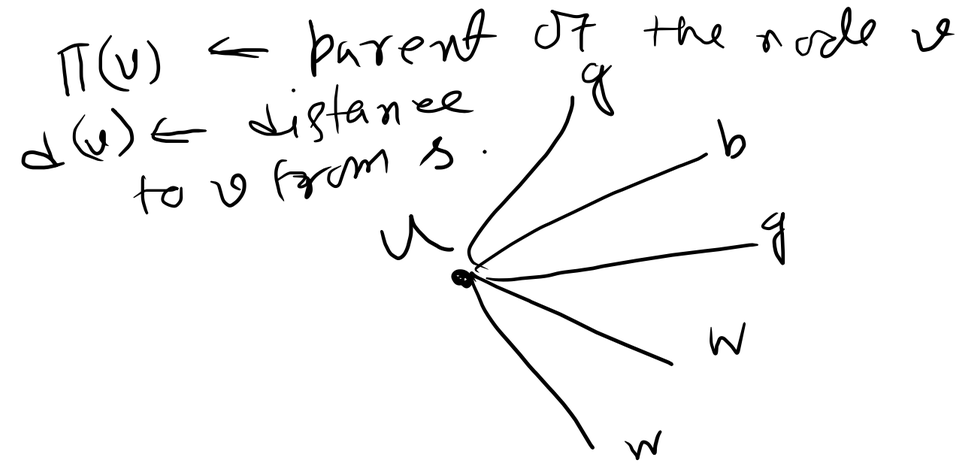
$\pi[v] = u$

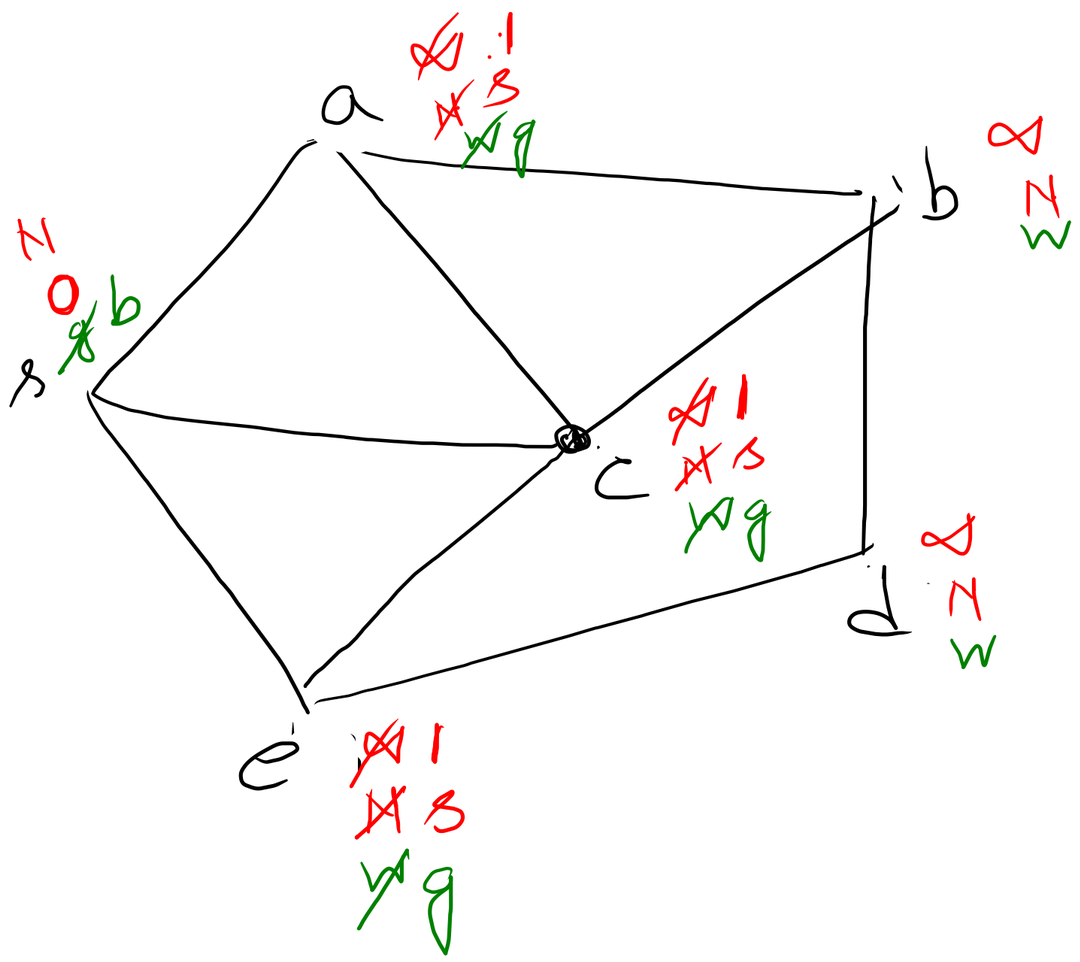
$d[v] = d[u] + 1$

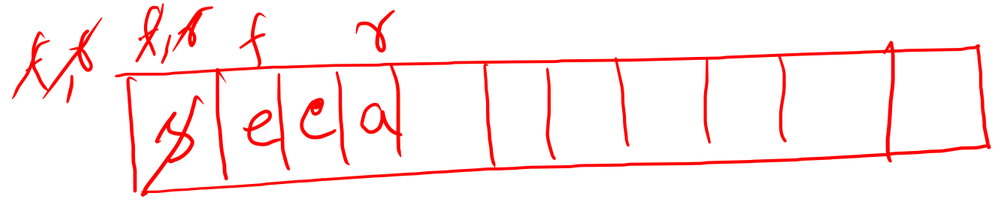
color(u) = black.

$\pi(u)$ ← parent of the node u

$d(u)$ ← distance to u from s.







$u = s$

$u = e$

H.W Finish this example.

Directed: only directed edges are present in the list.

Weighted: use an extra field in the node.