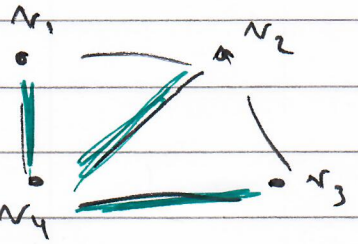
GRAPHS (NETWORKS)



PARALLEL EDGES / weighted edges / cost

$G = (V, E)$ — edges $u \leftrightarrow v$ — vertices — VERTEX

directed/undirected
length
capacity (flow)

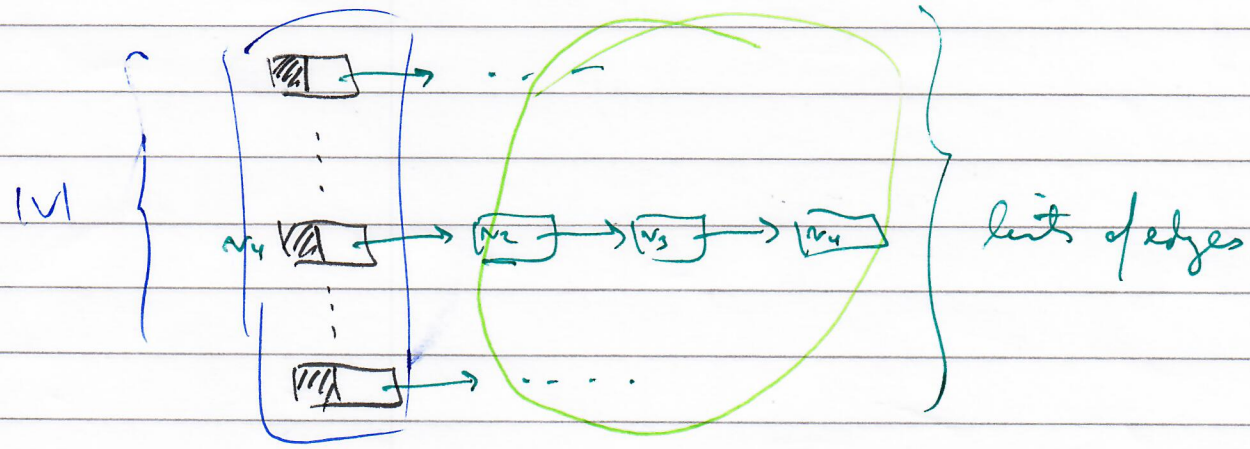
SIZE: $|V|$, $|E|$

Adjacency Structure

SPACE = $O(|V| + |E|)$

list of vertices

$O(|E|)$



Adjacency Matrix

$A = \begin{bmatrix} \cdots & j & \cdots \\ \vdots & & \\ i & \cdots & \\ \vdots & & \end{bmatrix}$ $|V|$ $|V|$

A_{ij} = some information about edge from $i \to j$
= true/false
= distance
⋮

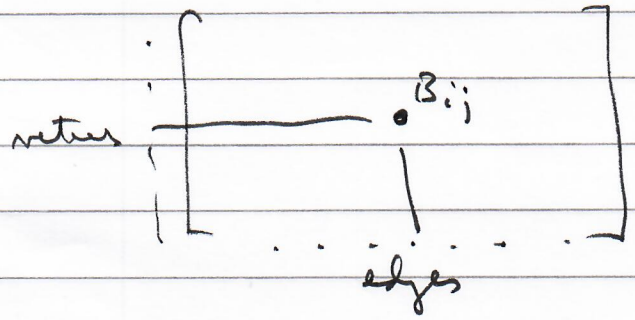
$O(|V|^2)$

$\begin{pmatrix} \cdots \end{pmatrix} \begin{pmatrix} \vdots \end{pmatrix} \begin{pmatrix} \cdot \end{pmatrix}$

Incidence Matrix

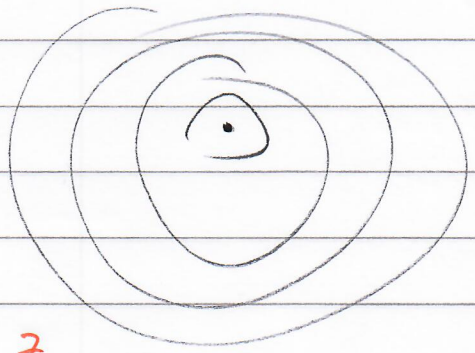
$B = [\quad]$ $\quad B_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ touches edge } j \\ 0 & \text{o.w.} \end{cases}$

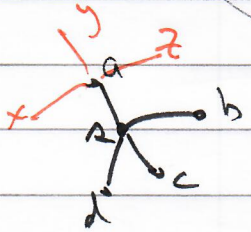
Space = $O(|V||E|)$

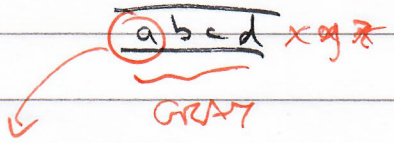


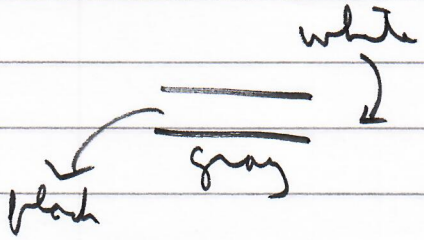
NP-hardness

QUEUE - BFS









```
function BFS(G, s)
1:  for all u ∈ V[G] − {s} do
2:    color[u] ← WHITE
3:    d[u] ← ∞
4:    π[u] ← NIL
5:  end for
6:  color[s] ← GRAY
7:  d[s] ← 0
8:  π[s] ← NIL
9:  Q ← {s}
10: while Q ≠ ∅ do
11:   u ← DEQUEUE (Q)
12:   for all v ∈ Adj[u] do
13:     if color[v] = WHITE then
14:       color[v] ← GRAY
15:       d[v] ← d[u] + 1
16:       π[v] ← u
17:       ENQUEUE (Q, v)
18:     end if
19:   end for
20:   color[u] ← BLACK
21: end while
```

O(|V|)

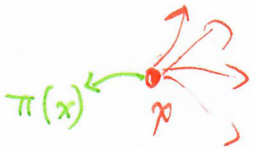
O(1)

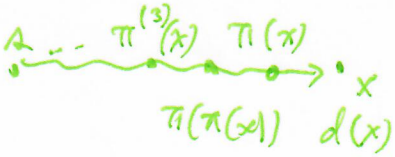
O(|E|)

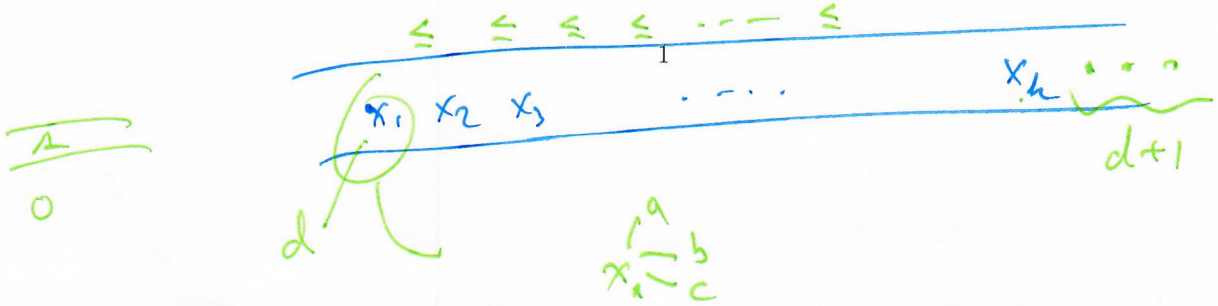
O(|V|)

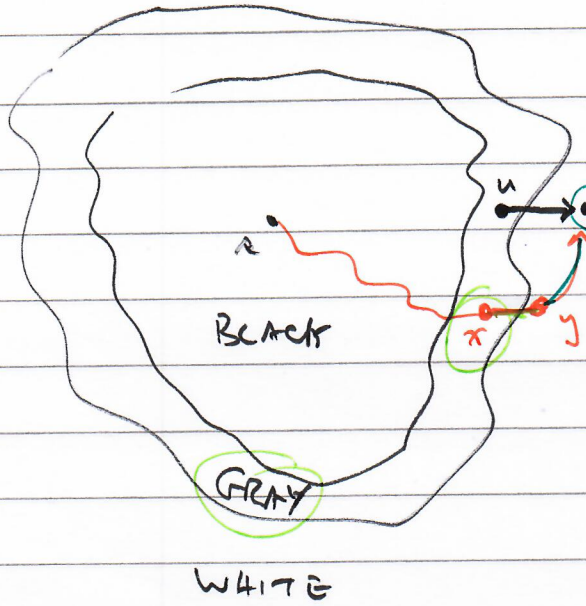
⇒ O(|V| + |E|)

Claim $d(v)$ = length shortest path from $s \rightsquigarrow v$ (edges)









$d(v) = d(u)+1$

length ≥ 0

$d(x) \geq d(u)$

$d(y) \geq d(x)+1$

$\geq d(u)+1$