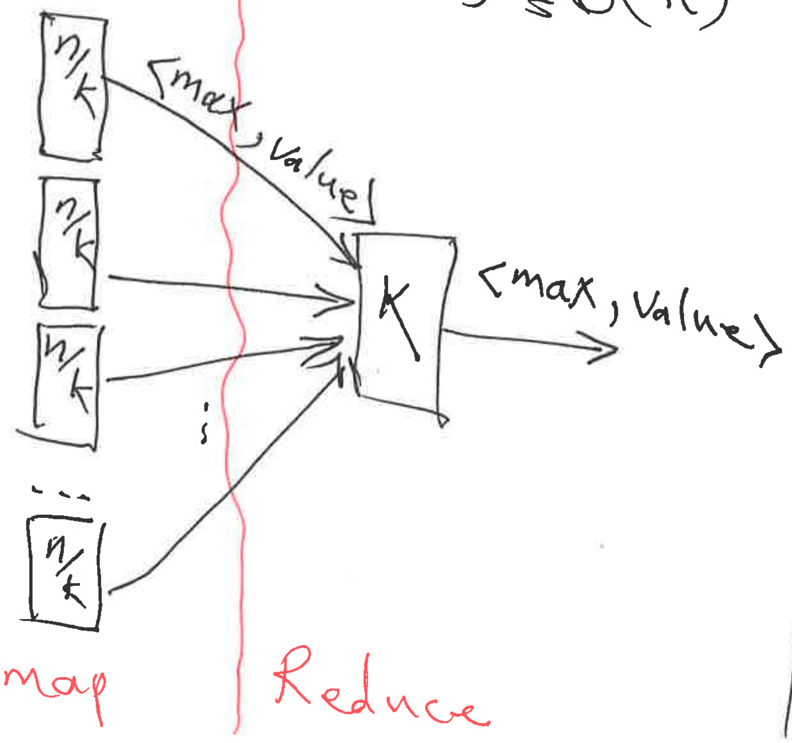
1

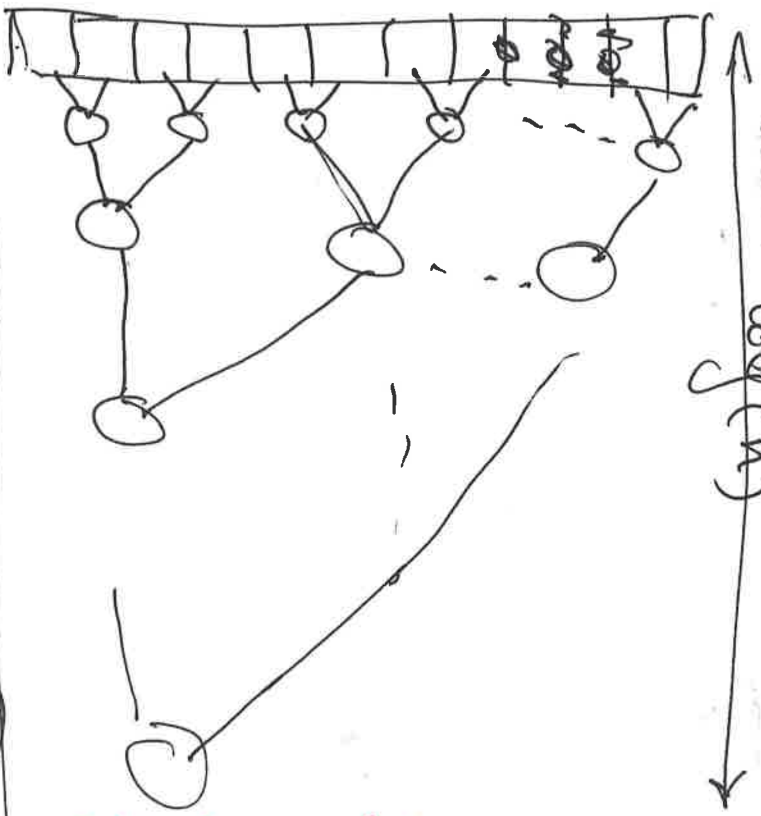
```
FindMax(A, low, high):
    if(low == high)
        return A[low]
    mid = ⌊(low+high)/2⌋
    ml = FindMax(A, low, mid)
    mr = FindMax(A, mid+1, high)
    return if(ml > mr) ml
           else mr
```

$T(n) = \underset{a}{2}\,T(n/\underset{b}{2}) + \Theta(n^{0})$ ← d

$a = 2 \quad > \quad 1 = b^{d}$

$T(n) = \Theta(n^{\log_2 2}) = \Theta(n)$





O(n) machines to Find Max in O(log n)

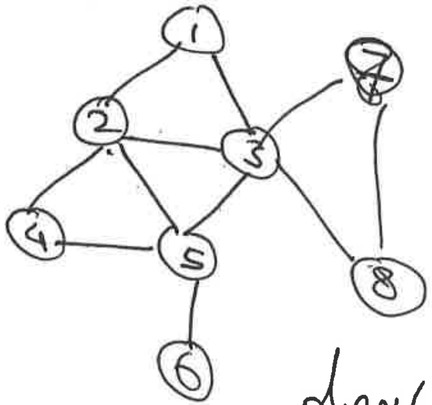
③ $G(V, E)$

undirected, unweighted

$v \in V$ is a node

$\langle v_i, v_j \rangle \in E$

n: # nodes, m: #Edges



$V = \{1, 2, \ldots 8\}$

$E = \{\langle 1,2 \rangle, \langle 1,3 \rangle, \ldots\}$

$n = 8$, $m = 11$

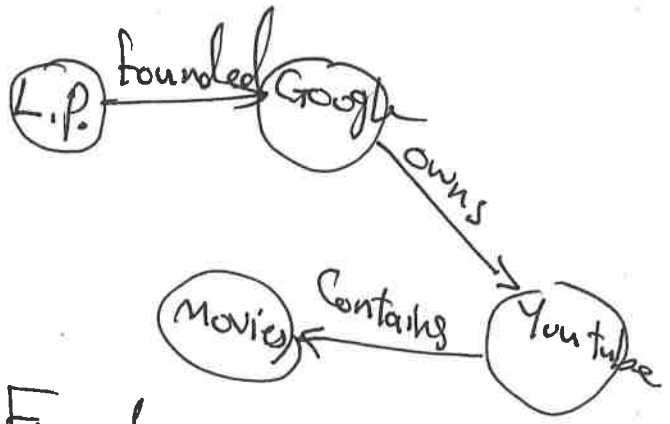
$deg(v_i) = |\{\langle v_i, v_j \rangle | E\}|$

$deg(6) = 1$, $deg(3) = 5$

Graphs
- undirected, unweighted
  - → facebook [friendship]
- directed, unweighted
  - → Twitter [followers]
- undirected, weighted
  - → A weight is assigned to an edge
  - → Maps.

...

Knowledge Graphs, AKA. Entity Graphs

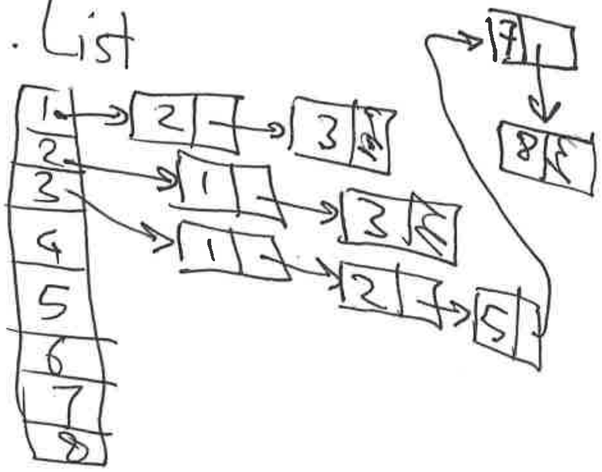


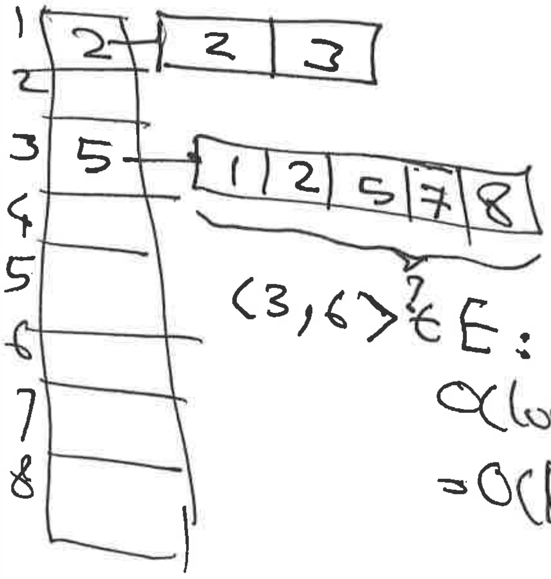
Freebase, DBPedia, wikipedia, ----

Graph Representation

$$\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
3 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
4 & & & & \cdots & & & & \\
5 & & & & & & & & \\
6 & & & & & & & & \\
7 & & & & & & & & \\
8 & & & & & & & &
\end{array}$$

Adj. List



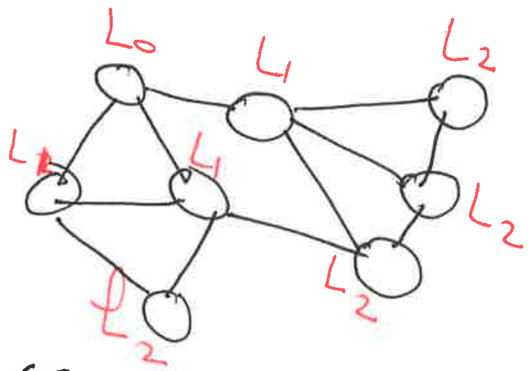


$\langle 3,6\rangle \overset{?}{\in} E$: B-Search

$O(\log(\deg(3)))$

$= O(\log n)$

Breadth First Search



BFS(G, S)

$L_0 = \{S\}$

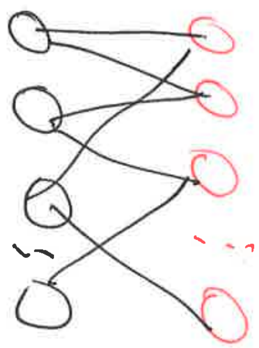
$L_1 = \{v_i \mid \langle v_i, S\rangle \in E\}$

L_2 = nodes that don't belong to L_0 and L_1 and have an edge to L_1

- - - -

L_{i+1} = nodes that don't belong to earlier layers and have edge to L_i

Bipartite Graph



every edge has 1 black and one red node.