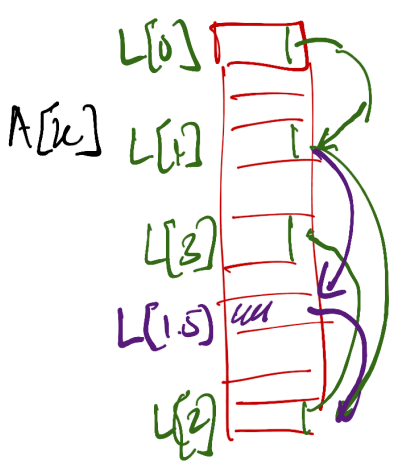
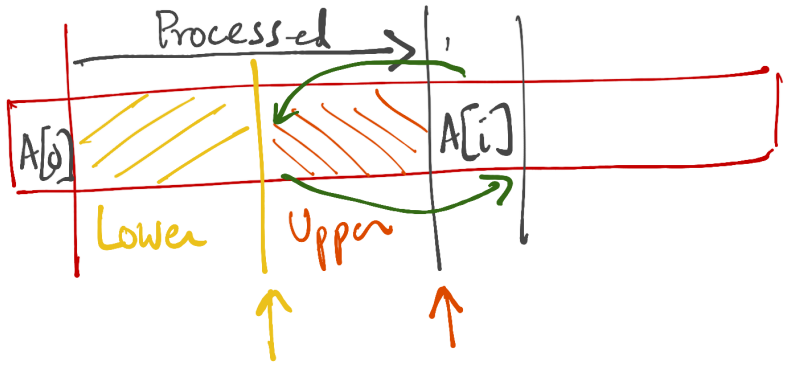
Array vs List



Insert between L[1] & L[2]

Finding L[k] takes O(k) time

Implementation [Assume distinct values]



A[i] > A[0]

A[i] < A[0]

Quicksort [CAR Hoare]

Pick some element - "Pivot"

A[0]

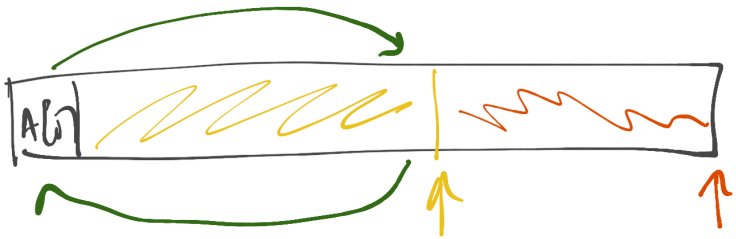
Lower = {A[i] | A[i] < A[0]}

Equal = {A[i] | A[i] = A[0]}

Upper = {A[i] | A[i] > A[0]}

Final List = [Sorted Lower] ++ [Equal] ++ [Sorted Upper]

Eventually



Good Case

$$T(N) = N + 2T(N/2)$$

$$\rightsquigarrow O(N \log N)$$

Worst Case

Either lower or Upper is empty

$$T(N) = N + T(N-1)$$

$$\rightsquigarrow O(N^2)$$ – like Insertion/Selection Sort

Can prove that

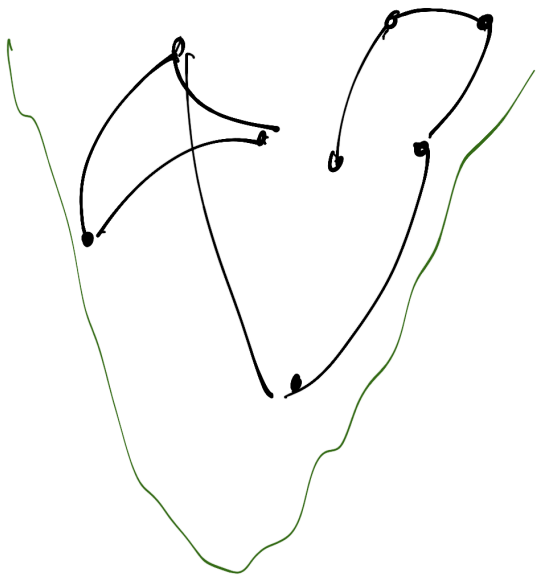
Average case is $O(N \log N)$

Alternatively

Pick pivot $A[i]$ randomly

Expected time is $O(N \log N)$

Airline Route Map



Graph

Nodes / Vertices

Edges — Directed (u,v)

Edges — Undirected $\{u,v\}$

Friends on a social network

Nodes – Users

Edges – Friend – Undirected

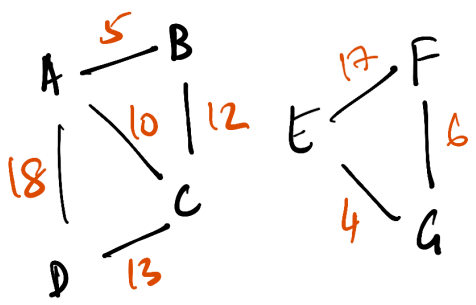
$G = (V, E)$

V is set of vertices

$E \subseteq V \times V$

+ Weights "Cost" : $E \to \mathbb{R}$

Mechanism to process graphs algorithmically



Can I go from

B to D?

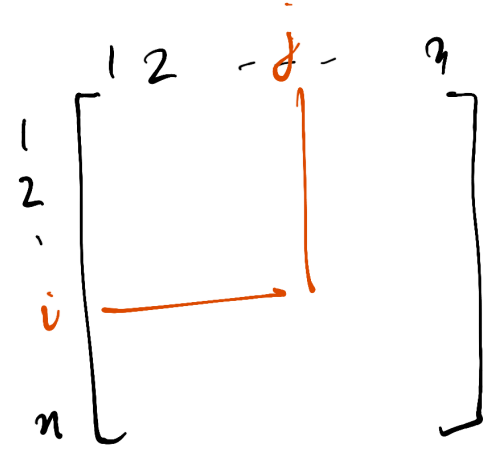
E to D?

Need a representation for a graph

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

$E \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$

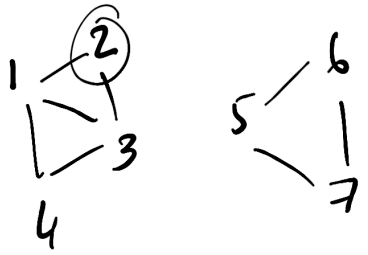
A =

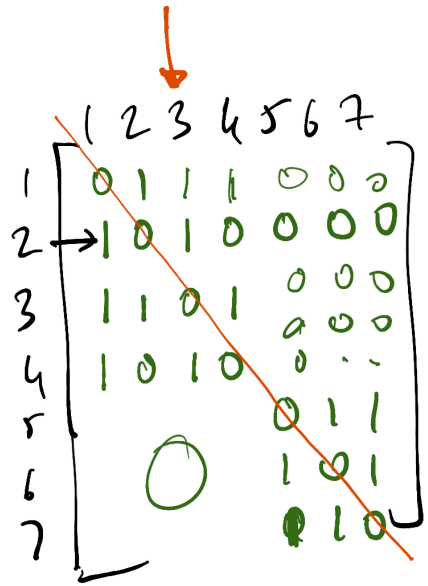


A[i][j] = 1 iff $(i, j) \in E$

Adjancency Matrix

Neighbours of 2?





From where can I reach 3 in one step?

Both take $O(n)$ time

$G = (V, E)$

$|V| = n$

$|E| = m$

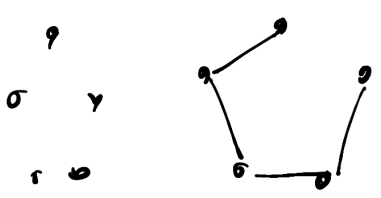
Bounds on m?

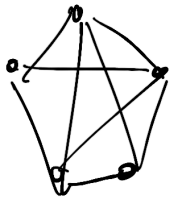
Max

Undirected $\binom{n}{2}$

Directed $n(n-1)$

Min connected $n-1$





m is between $O(n)$ and $O(n^2)$

sparse dense

Adj Matrix is $O(n^2)$ in size, always

Finding neighbours takes $O(n)$, always

degree(v) = # neighbours of v

Can we represent sparse graphs better?

Adjacency list

$1 \to \{2,3,4\}$

$2 \to \{1,3\}$

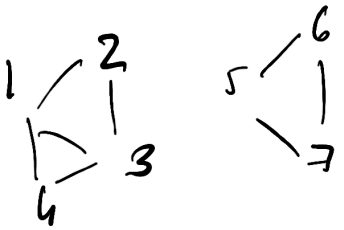
$3 \to \{1,2,4\}$

$4 \to \{1,3\}$

$5 \to \{6,7\}$

$6 \to \{5,7\}$

$7 \to \{5,6\}$



Neighbours of i

List for i

Degree(i)

$1 \to \{2,3,4\}$

$2 \to \{3\}$

$3 \to \{\}$

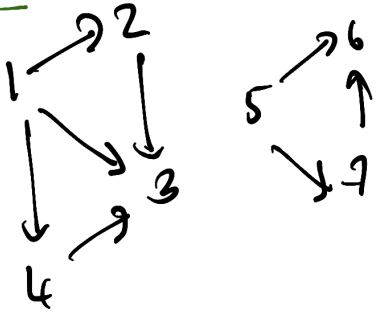
$4 \to \{3\}$

$5 \to \{6,7\}$

$6 \to \{\}$

$7 \to \{6\}$

Directed



Who point to 3?

Scan all lists - $O(m)$

q questions $\to O(qm)$

$1 \to \{2,3,4\}$ ✓

$2 \to \{5\}$ ✓

$3 \to \{\}$ ✓

$4 \to \{3\}$ ✓

$5 \to \{6,7\}$ ✓

$6 \to \{\}$

$7 \to \{6\}$ ✓

Reverse edge list

1 $\{\}$

2 $\to \{1\}$

3 $\to \{1,2,4\}$

4 $\to \{1\}$

5 $\{\}$

6 $\to \{5,6\}$

7 $-\{7\}$

Problems?

1. Reachability

Can I get from s to t?

Where can I go in 1 step?

2 steps? ---

Breadth first search (BFS)

Weighted graphs?

Each edge has a weight: $c: E \to \mathbb{R}$

$c(e) \neq 0$

Representation?

Adj. Matrix $A[i][j] = c((i,j))$

Adj list $j \to \{(i, c((j,i))), \ldots$