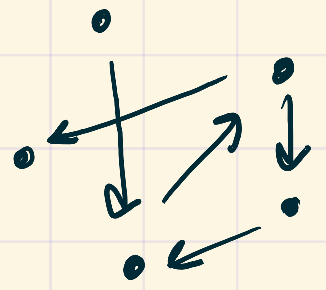
MATH 2301 12 August 2020

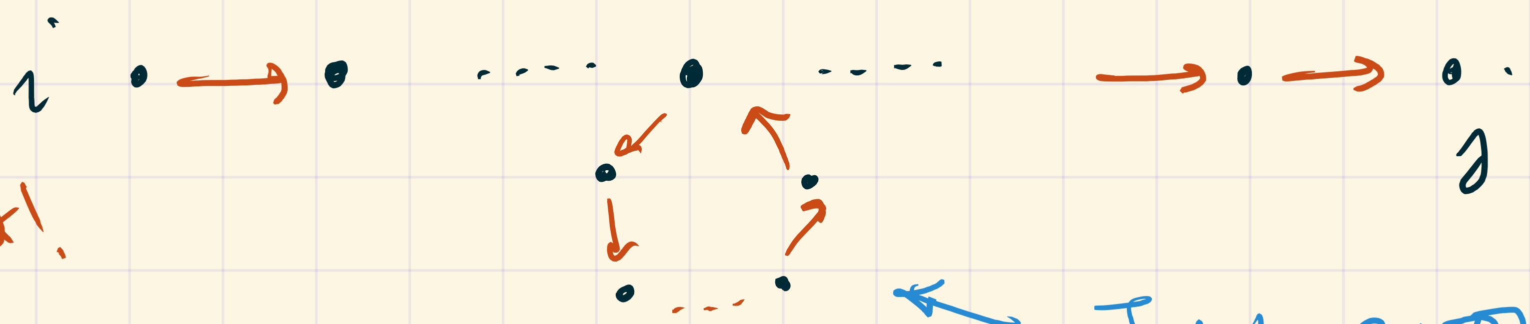
* **Recap**: Adjacency matrices & related calculations



* **Recall**:

(1) If A = adjacency matrix of a graph (V,E), then the $(i,j)^{th}$ entry of A^k is the number of paths of length k from i to j.

(2) If $i \neq j$ and if there is at least one path from i to j, then the shortest such path cannot repeat any vertex:



Just avoid this loop to make path shorter!

Important!

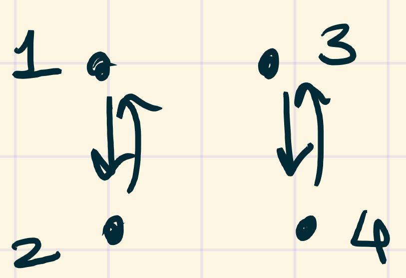
Since there are n vertices, the shortest path from i to j has length $\leq (n-1)$ if $i \neq j$

* If $i=j$ the shortest non-zero path from $i \to j$ has length $\leq n$.

Theorem: There is a path from i to j of some length if and only if the $(i,j)^{th}$ entry of

$$A + A^2 + \cdots + A^n \text{ is non-zero.}$$

E.g:



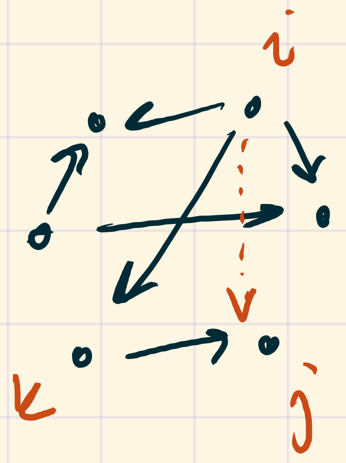
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I_4$$

$$A^3 = A \qquad A^4 = A^2$$

$$A + A^2 + A^3 + A^4 = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix} \Rightarrow \text{No paths from 1 to 4, for example!}$$

* Transitive closure :

Given a relation R on a set S, look at the graph (S, R).



To take transitive closure of R,

① Add the edge $i \to j$ whenever you see an edge $i \to k$ & $k \to j$

② Repeat until no new edges show up.

In other words, we're adding an edge $i \to j$, whenever there is a (positive-length) path from i to j:

$$i \to \cdots \to j$$

---

* Use a simpler variant of matrix multiplication

Boolean product

- Defined on matrices of 0s and 1s.
- Like the usual matrix product, except we replace "$\times$" with "and" = "$\wedge$", "$+$" with "or" = "$\vee$"

$0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0$

$1 \wedge 1 = 1$

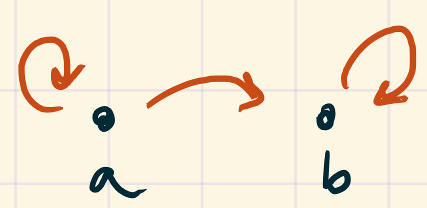
$0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$

$0 \vee 0 = 0$

"1" stands for "true"

"0" stands for "false"

Example : $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$



$A * A$ = Boolean matrix square $= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

$$= \begin{bmatrix} (1\wedge 1) \vee (1 \wedge 0) & (1 \wedge 1) \vee (1 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) \end{bmatrix} = \begin{bmatrix} 1 \vee 0 & 1 \vee 1 \\ 0 \vee 0 & 0 \vee 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{= A}$$

Contrast this with $A^2 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$

$$A^{(*2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

We'll say that $A^{(*k)}$ is the k-fold Boolean product.

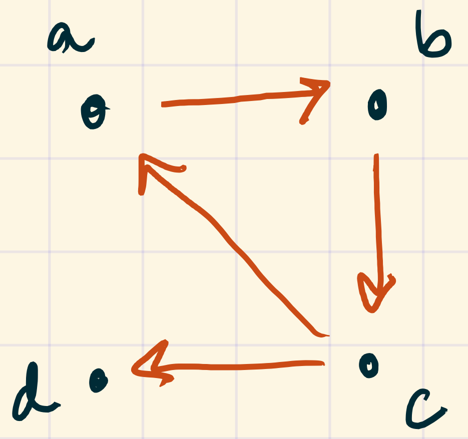
<u>Note</u> :

The $(i,j)^{th}$ entry of $A^{(*k)}$ equals 1 if and only if there is a path of <u>length k</u> from i to j.

<u>Theorem</u> : The matrix of the transitive closure is given by : $\underline{\underline{A}} \vee \underline{\underline{A}}^{*2} \vee \underline{\underline{A}}^{*3} \cdots \vee \underline{\underline{A}}^{*n}$ (n = # of vertices)

is there a path of length 3 ?

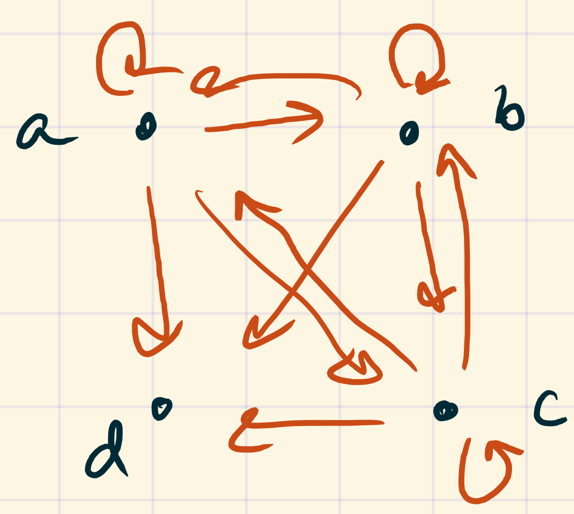
<u>Example</u> :



$$A = \begin{bmatrix} \boxed{0} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A^{*2} = \begin{bmatrix} \boxed{0} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A^{*3} = \begin{bmatrix} \boxed{1} & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A^{*4} = \begin{bmatrix} \boxed{0} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A \vee A^{*2} \vee A^{*3} \vee A^{*4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



Transitive closure of old graph.

<u>Note</u>: It's quicker to compute matrix powers (normal or Boolean) by <u>repeated squaring</u>:

i.e $A^4 = A \cdot A \cdot A \cdot A$ ← 3 matrix products

$= (A^2)^2$ ← 2 matrix products!

$A^7 = (A^2)^2 \cdot A^2 \cdot A$ ← 5 matrix products

vs 6 you'd usually need...

<u>Rmk</u>: Fast matrix multiplication is an active area of research!