MATH 2301 — 11/08/2023

* Last time: Transitive closure of R
  = smallest superset $R'$ of R (i.e $R \subseteq R'$) such that $R'$ is transitive.

Adjacency matrix of transitive closure:

If A = adj. matrix of a graph with n vertices, take $(A + A^2 + \cdots + A^n) = B$, and change B so that

- all non-zero (positive) entries become 1
- all zero entries kept as-is.

The changed B is the adj. matrix of the transitive closure

* Today: Do the same w/ Boolean arithmetic

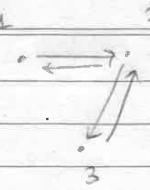
Operations $\vee$ = OR, $\wedge$ = AND } binary operations on $\{0, 1\}$ ("FALSE", "TRUE")

| | |
|---|---|
| $0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$ <br> $0 \vee 0 = 0$ | $1 \wedge 1 = 1$ <br> $1 \wedge 0 = 0 \wedge 1 = 0 \wedge 0 = 0$ |
| (like an addition) | (like a multiplication) |

* Boolean matrix product: ~~as~~ like usual matrix product, but replace every instance of + with $\vee$ and $\times$ or $\cdot$ with $\wedge$



Eg.




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

$A^{*2}$ = 2nd Boolean power
($*$ = Boolean matrix product)

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

$(A*A)_{(2,2)}$ = Boolean dot product of $[1, 0, 1]$ with $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

$= (1 \wedge 1) \vee (0 \wedge 0) \vee (1 \wedge 1) = 1 \vee 0 \vee 1 = 1$

($\wedge$ replaces $\times$; $\vee$ replaces $+$)

Theorem / consequence: Let A be the adj matrix of a graph with n vertices.
The adjacency matrix of the transitive closure of A is:

$$(A \vee A^{*2} \vee A^{*3} \vee \cdots \vee A^{*n})$$

* Aside: Given a relation R, you can define the

① reflexive closure

② symmetric closure

(Remember, the <blah> closure of R is the smallest relation R' such that $R \subseteq R'$ and R' is <blah>)

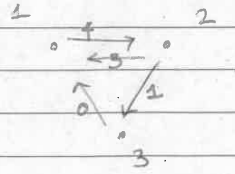
Q: What about anti-symmetric closure?
(This doesn't make sense → can you figure out why?)

Q: How to modify the adjacency matrix to get the adj matrix of the reflexive/symmetric closure?

---

* Weighted graphs

A weighted graph is some $G = (V, E)$, together with a weight (a number) for each edge.




(Think of weights as either a cost or a capacity for each edge.)

→ Can construct the weighted adjacency matrix

$$W = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} * & 4 & * \\ 3 & * & 1 \\ 0 & * & * \end{pmatrix} \end{array}$$

$(i,j)^{th}$ entry = weight of the edge from $i$ to $j$

$*$ if no edge $i \to j$



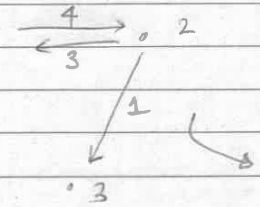
What should $*$ be?
→ Depends on the application.

For us, we focus on regarding the weights as costs.

If there is no edge from $i$ to $j$ ⇒ "the cost is too high to pay"

→ We'll use $\infty$ to denote this, except at $i=j$

* $\infty$ is not a number, it is a symbol, we'll see how to handle it in calculations.

Ex 


Find shortest-cost (cheapest) path from $i$ to $j$

$$W = \begin{pmatrix} 0 & 4 & \infty \\ 3 & 0 & 1 \\ \infty & \infty & 0 \end{pmatrix}$$