Dynamic All-Pairs Reachability

Reachability	V/s Connectivity
$x \xrightarrow{\text{reachable}} y$	$x \xleftrightarrow{\text{connected}} y$
In directed "G" there is path from x to y	In undirected "G" x and y are connected

Static Scenario : Time to compute all-pair reachability information (transitive closure) is $O(\min\{(mn), \underbrace{(n^{\omega} \log n)}_{\text{repeated matrix multiplication}}\})$

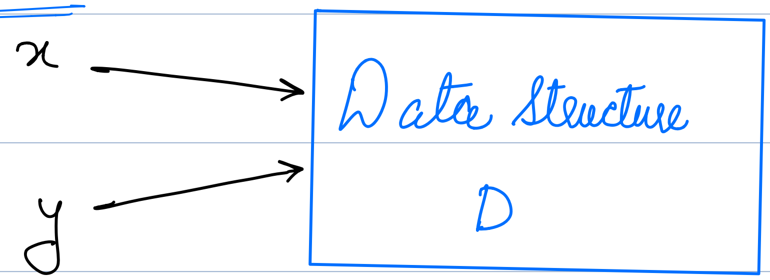
(graph is not changing)

$\omega \leq 2.38$

ω = constant of matrix multiplication

Question: How to dynamically maintain transitive closure in fully-dynamic setting?

Goal:



Answer if y reachable from x in $O(1)$ time

Edge addition / deletion

Time to update D on edge addition / deletion $= O(n^2)$

Our algorithms will work only for $\underbrace{\text{DAGs}}_{}$ Directed acyclic graph

Key Ideas:

Modulo prime for working with large nos.

Prime Number Theorem: The number of primes less than n, "$\pi(n)$" is $\Theta(n/\log_e n)$.

$$\frac{7}{8}\frac{n}{\ln n} \leq \pi(n) \leq \frac{9}{8}\frac{n}{\ln n}$$

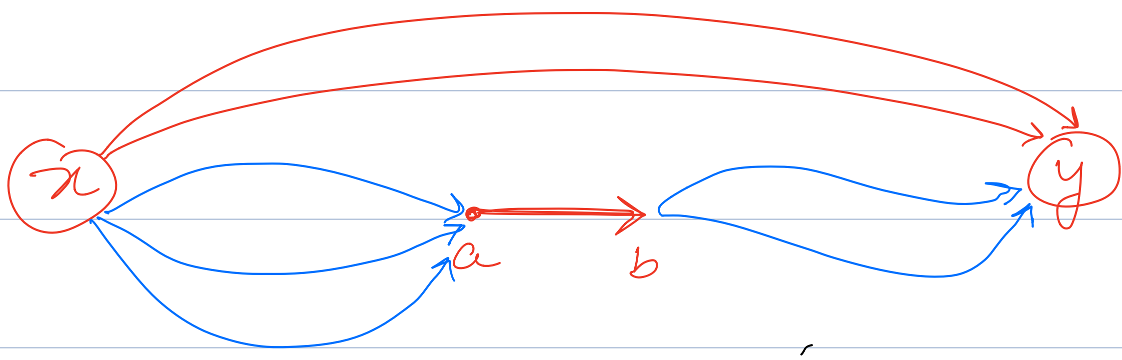
Algorithm

$\forall\, x, y \in (V \times V)$, $C_t(x,y) =$ Total number of paths from x to y in G_t. (i.e at time "t")

- On addition of edge (a,b) to G:

For all $x, y \in (V \times V)$,

$$C_{t+1}(x,y) = \underbrace{C_t(x,y)}_{2} + C_t\underbrace{(x,a)}_{3} \cdot C_t(\underbrace{b,y}_{2})$$



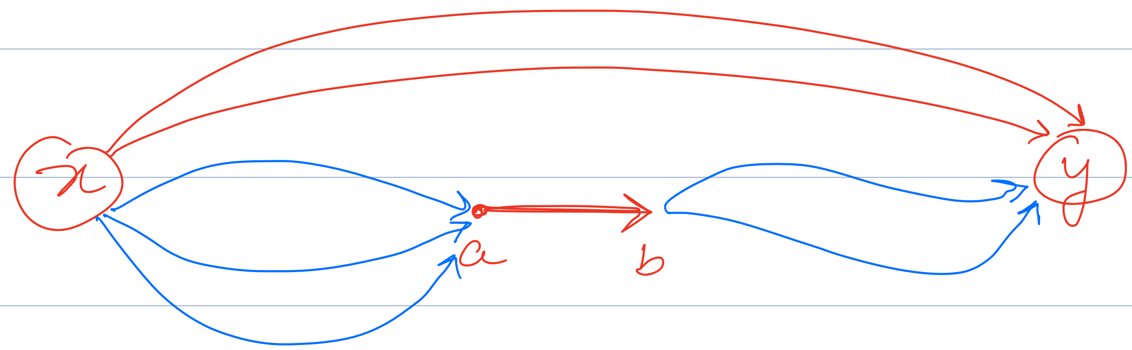
Question: Where are we using acyclicity?

Proof: H.W.

- On deletion of edge (a,b) from G:

For all $x,y \in (V \times V)$,

$$C_{t+1}(x,y) = \underbrace{C_t(x,y)}_{8} - C_t(\underbrace{x,a}_{3}) \cdot C_t(\underbrace{b,y}_{2})$$

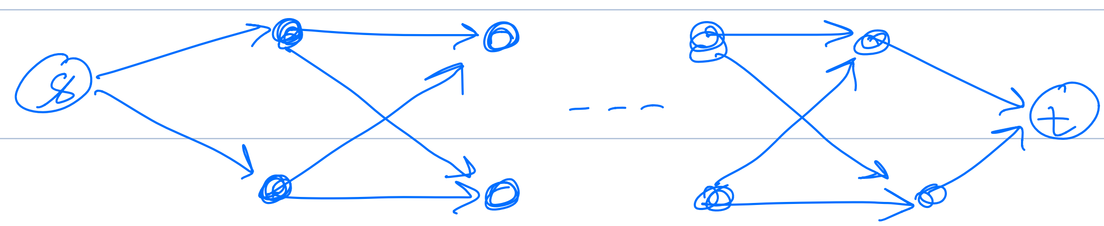


Question: What is time to update $C(x,y)$, for a given pair (x,y)?

Answer: $\log_2(C_{t+1}(x,y)) + \log_2(C_t(x,y))$.

Now, $C(x,y)$ can be as large as 2^n.

Eg:



So, time to update single pair $= \Omega(n)$

$\square$

But notice there are n^2 pairs

Suppose the sequence of updates is n^k,
(i.e. $1 \leq t \leq n^k$), then $C_t(x,y)$ can take n^{k+2}
distinct values. (Here $n^{k+2} <<< 2^n$).

$$p = \text{random prime in range } [n^{2k}, n^{3k}]$$

$$\text{define } \underbrace{\tilde{C}_t(x,y)} = C_t(x,y) \bmod p$$

This uses just $\underbrace{3k \log n}$ bits as $p \leq n^{3k}$.

Some Identities

$$(a+b) \bmod p = (a \bmod p + b \bmod p) \bmod p$$

$$(a-b) \bmod p = (a \bmod p - b \bmod p) \bmod p$$

$Z_p = \{0, 1, 2, \ldots, p-1\}$

$Z_p^* = \{1, 2, \ldots, p-1\}$

We use modular airthmetic (mod p) so as to work with smaller numbers.

__Addition__

$(17 + 13) \bmod 7$ $\quad (p=7)$

$= (17 \bmod 7 + 13 \bmod 7) \bmod 7$

$= (3 + 6) \bmod 7$

$= 2$

__Multiplication__

$(17 * 13) \bmod 7$

$= ((17 \bmod 7) * (13 \bmod 7)) \bmod 7$

$= (3 * 6) \bmod 7$

$= 4$

Why we choose "p" to be prime ?

Suppose $p = 20$, $a = 25$, $b = 24$

$\rightarrow$ $a \bmod p = 25 \bmod 20 = 5 \in Z_{20}$

$\rightarrow$ $b \bmod p = 24 \bmod 20 = 4 \in Z_{20}$

$(5 \cdot 4) \bmod 20 = 0$

5 & $4 \in Z_{20}$ are non-zero but $(5 * 4) \bmod 20 = 0$

Product of non-zero elements in this case has become zero, however product should be zero when at least one of them is zero.

H.W. If p is a prime, and $a, b \in Z_p \setminus \{0\}$, then $(a \cdot b) \bmod p \in Z_p \setminus \{0\}$.

Hint: $x \bmod p = 0$ iff $\frac{x}{p}$ is integer

- By Prime number Thm: At least $\frac{n^{2k}}{k \cdot \log n}$ primes in $[n^{2k}, n^{3k}]$.

- Note: $C_t(x,y) \leq (n!)$. (n factorial)

- Number of primes in $[n^{2k}, n^{3k}]$ that divide $C_t(x,y) \leq \log_2(C_t(x,y)) \leq n \log n$

$$\sum_{\substack{1 \leq t \leq n^k \\ (x,y) \in V \times V}} \text{No. of prime factors of } C_t(x,y) \leq n^k \cdot n^2 \cdot n \log n \leq n^{k+4}$$

$$\Pr\left[\begin{array}{l} p \text{ divides } C_t(x,y) \\ \text{for some } 1 \leq t \leq n^k, \\ x \in V,\ y \in V \end{array}\right] \leq \frac{n^{k+4}}{\frac{n^{2k}}{\cancel{k \cdot \log n}}} \leq \frac{1}{n^{k-6}}$$

$\boxed{\text{We can take } k = 8}$

We are working with $3k(\log_2 n)$ bits.

Note that word size is $\Omega(\log_2 n)$, as these many bits are needed to store labels of graph indices

Thus we will be using at most "3k" words to reach $(3k)\log_2(n)$ bits.

$k = O(1)$ (recall $k=8$ suffices)

So we are requiring $8 \cdot 3 = 24 = O(1)$ words to store $\tilde{C}_t(x,y)$.

Time to find $\tilde{C}_{t+1}(x,y)$, for all (x,y) pairs is thus just $O(n^2)$.

We say $x \rightsquigarrow y$ exists in G_t iff $\tilde{C}_t(x,y)$ is non zero.

The answers are correct with high probability.