14 Oct 2019

Asymptotics

① $f \in \Theta(\log_2 n) \iff f \in \Theta(\log_3 n)$

$\iff f \in \Theta(\log_k n)$, k = const.

② $2 \leq 3$ $\quad f \in \Theta(n^2) \Rightarrow f \in O(n^3)$

$f \notin \Theta(n^3)$

$a \leq b$ $\quad f \in \Theta(n^a) \Rightarrow f \in O(n^b)$

$f \notin \Theta(n^b)$

③ ordering of common asymptotics

$\Theta(1) \lneq \Theta(\log n) \lneq \Theta(n) \lneq \Theta(n \log n)$

"constant" — logarithmic — linear

$\lneq \Theta(n^2)$ ... poly...

quadratic

$\lneq \Theta(2^n)$

exponential

④ Fun Fact $\Theta(n^\omega)$ is "matrix mult. time"

$\Theta(n^2) \leq O(n^\omega) \leq \Theta(n^3)$

currently $\omega = 2.3737\ldots$

⑤ The Ackerman fcn

$A: \mathbb{R} \to \mathbb{R}$ is the ackerman fcn...

intuitively, this is really fast growing.

$\alpha: \mathbb{R} \to \mathbb{R}$

$x \mapsto A^{-1}(x)$ is the inverse Ackerman fcn

grows very slow

in practice, will be constant.

Problem: Given a graph, want to know
① how many connected components?
② are vertices x & y connected?

Examples:

- airline flights
  - nodes = cities
  - flight paths btwn cities = edges
  - can I get from city A to city B?
- galaxies
  - nodes: stars w/in one galaxy
  - edges: distance btwn stars

→ If I threshold, how many clusters of stars do I see?

- Netflix:
  - nodes = shows
  - edges = connected if person A watches show

↳ set of graphs $G = \{G_A\}_{A=\text{person}}$

- nodes = people, rep as a vector (show 1 count, show 2 count, ...)
- edges = weighted dist b/c 2 vectors.

- web as a graph.
  Q: Is there 1 spider or 2?

Union-find data structure

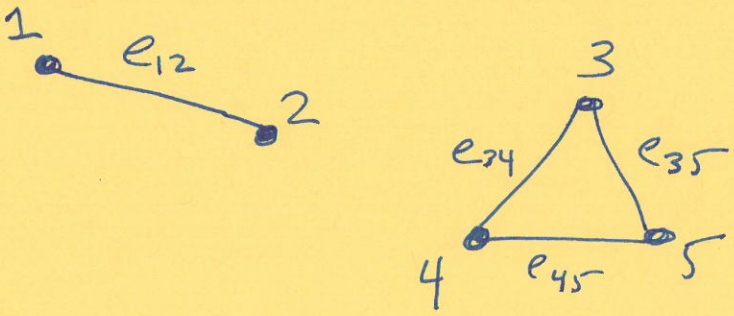
supports 3 operations

① makeNode → can be done as many times as needed, but only before other operations start

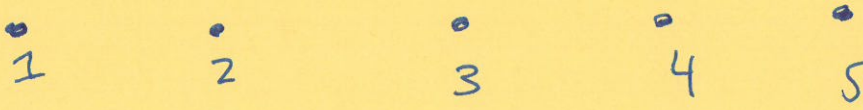
② UNION (a, b) → joins a's component w/ b's component, if not already the same component.

③ FIND (a) → finds the unique label corresponding to a's component

example:

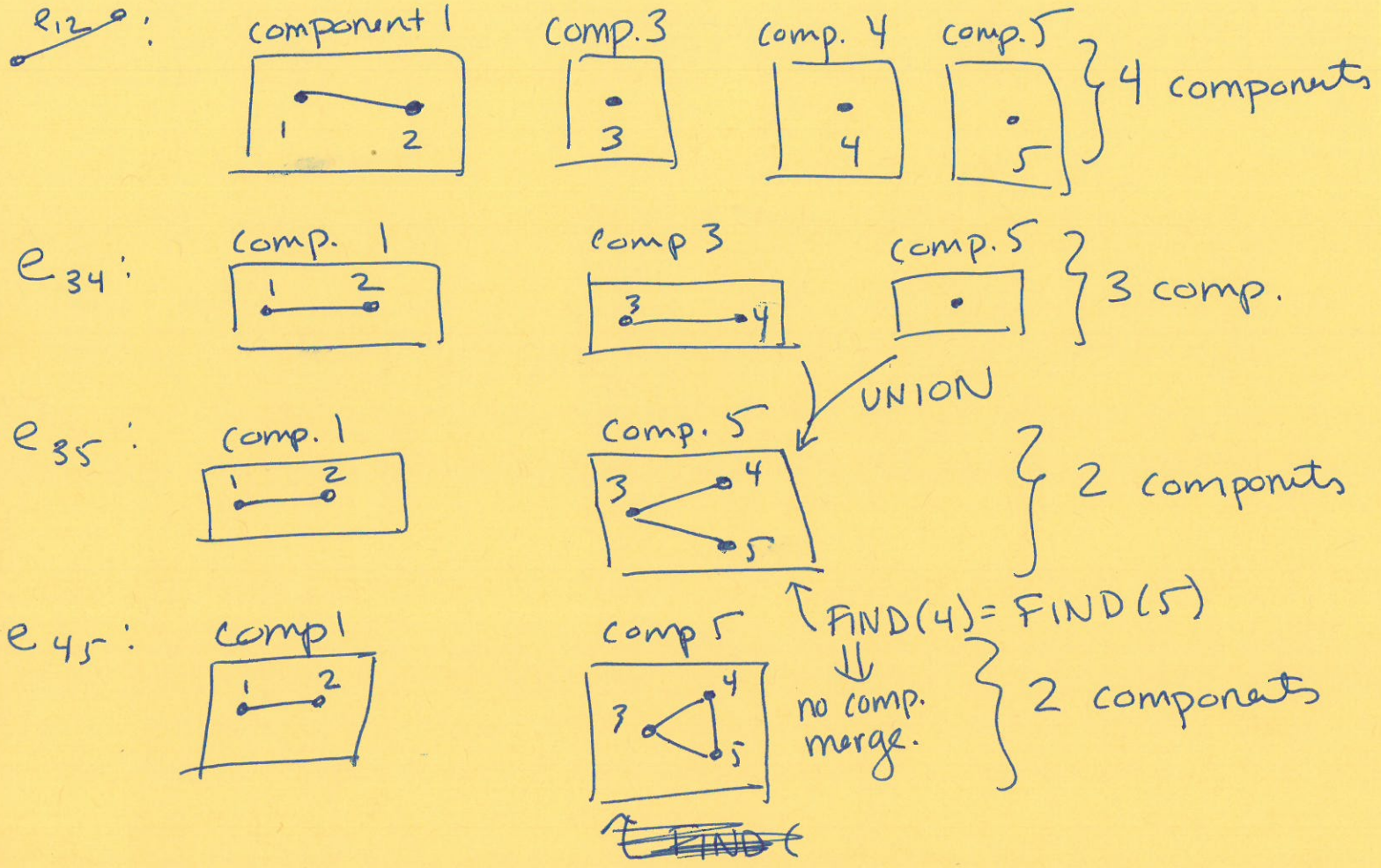


① create one component per node



5 components

② go through all the edges.



Underlying q: M union or find operations.

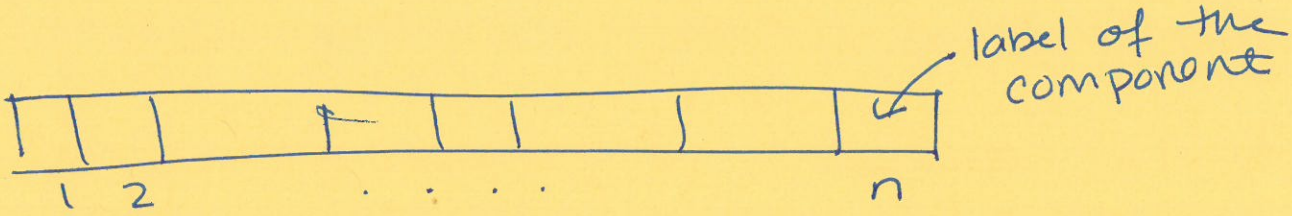
How long will this take?

First Attempt: Quick Find

- you give me index of vert, I quickly give you the component label

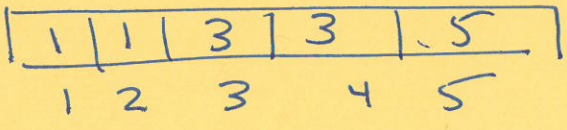
DS = an array (hash table can work too, w/ average case analysis)

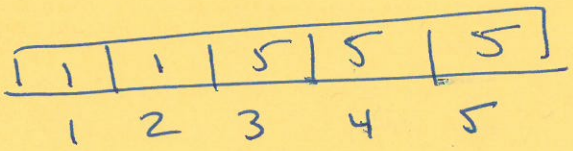
note: Let's assume vert ~~are~~ have ids 1 ... n



⇒ FIND is $\Theta(1)$.

⇒ UNION is $\Theta(n)$.





So, after M operations, could cost $O(Mn)$