18 Oct 2019

UNION-FIND

QUICK UNION:

→ each vertex saves "parent" nod & to find the component label, go up tree until you find the root

UNION (given 2 root nodes): $\Theta(1)$

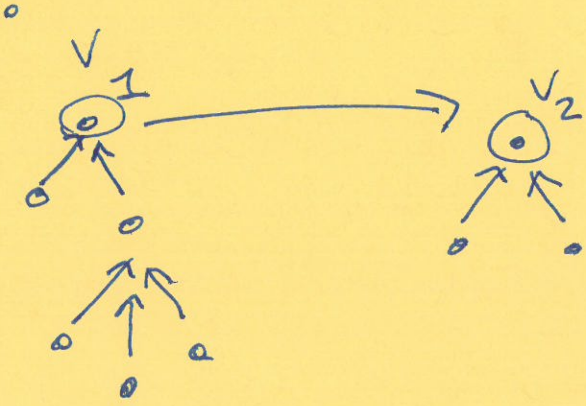
FIND (to find the root node): $\Theta(n)$

UNION (given 2 arb. nodes)

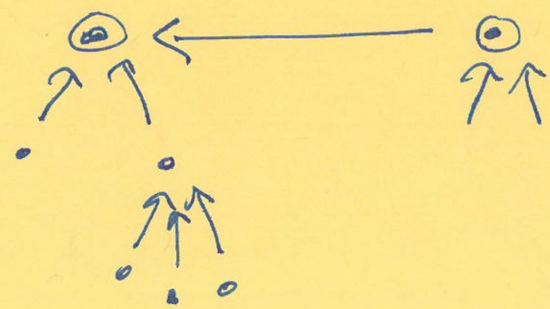
$= 2$ FINDS $+$ 1 UNION FROM ROOT NODES

$= \Theta(n) + \Theta(n) + \Theta(1) = \Theta(n)$

IMPROVEMENT-1:



OR



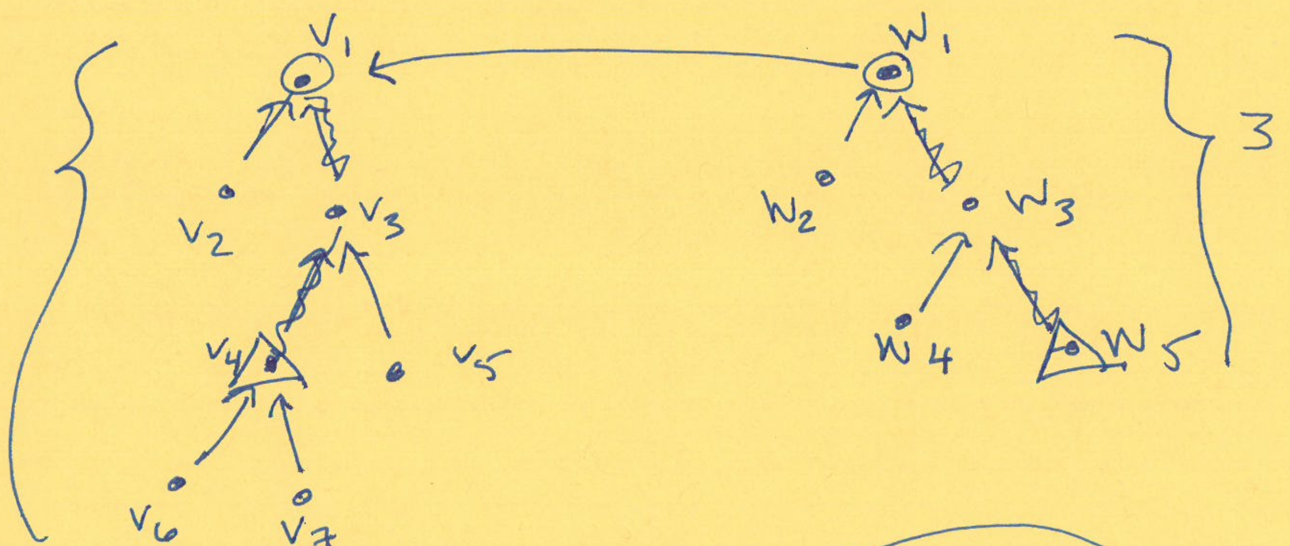
} preferred way

- if they are the same size, then depth increases by 1
- if they aren't, depth does not change.

$\Longrightarrow$ FIND $\Theta(\log n)$

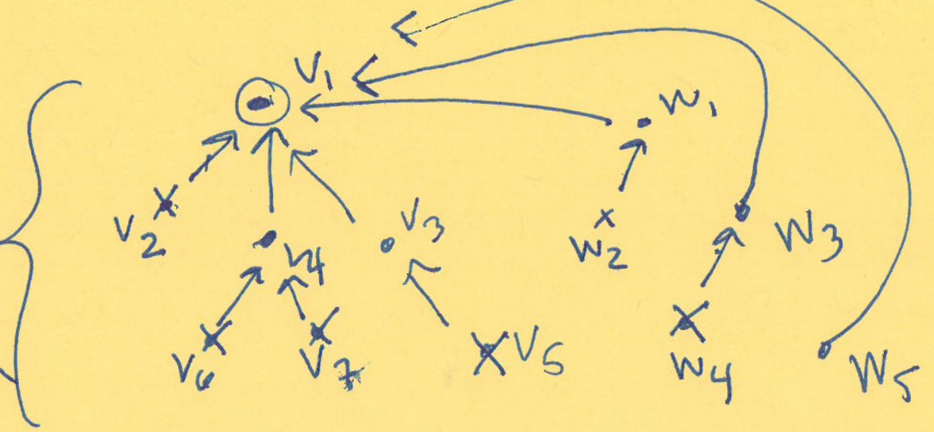
UNION $= 2 \cdot$ FIND $+ \Theta(1) = \Theta(\log n)$

IMPROVEMENT #2: Path Compression



depth: $3 \leq 4$

pretending path compression did not happen



Still merge on "depth potential"

M operations of either UNION / FIND, then the runtime

$\Theta(\alpha_n \cdot M)$

mult.

the inverse Ackermann fcn

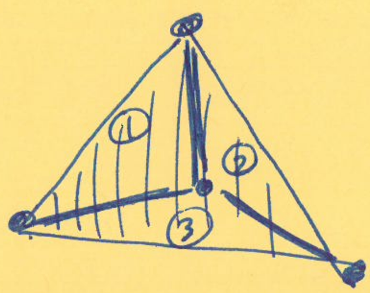
nearly linear in M.

Triangulations (Planar)

Given $P \subseteq \mathbb{R}^2$, $|P| < \infty$

Want: to use that point set to ~~triangul~~ decompose $CH(P)$ into triangles, edges, + vertices

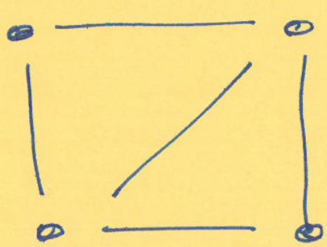
e.g.



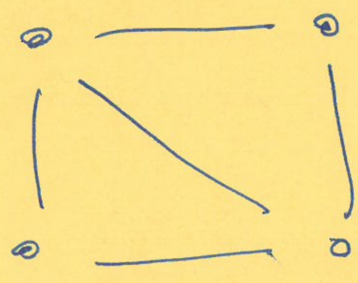
$CH(P) :=$ the intersection of all half-planes containing P

$:=$ the smallest convex shape containing P

e.g.,



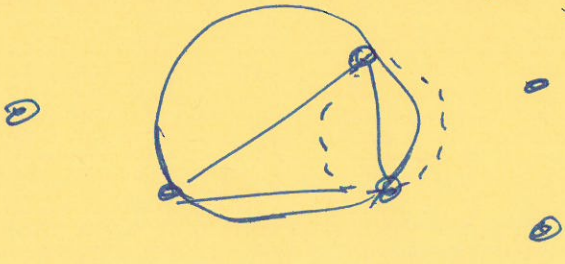
OR

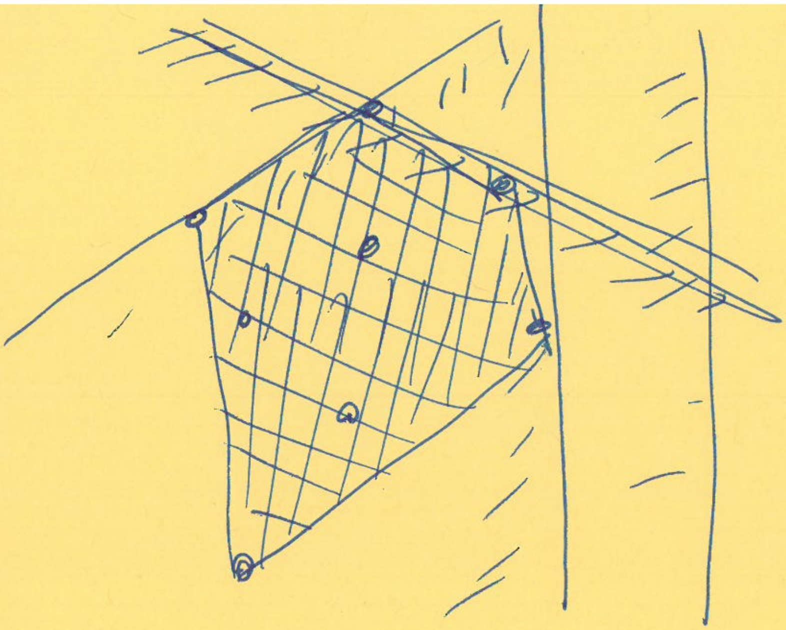


Special Δulation: Delaunay Δulation.

Def!: • for every triangle, the circumscribing circle is empty of points in P

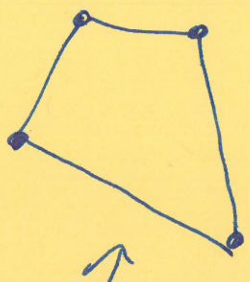
Def 2: Also, every edge has the empty circle property too.



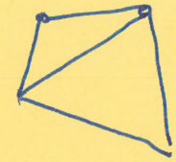


Q: What if I am given a triangulation that is not a Delaunay Δulation. Can I fix it in order to obtain the Delaunay Δulation?

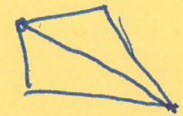
note: there is always a Delaunay Δulation of any set of pts. And it is unique, assuming general position of the vertices.



2 triangulations:



✓Delaunay



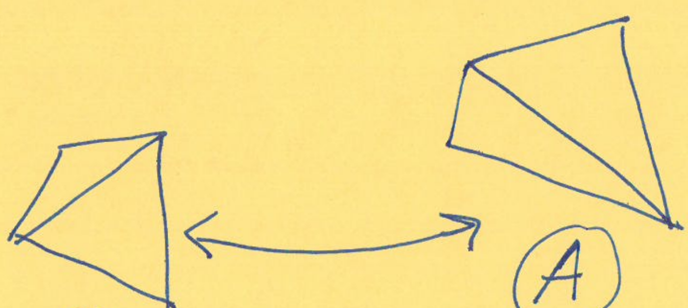
NOT Del.

the 2 ways that 4 verts can be configured "in general position"

the only Δulation

✓Delaunay

Edge flip operation: given an edge whose 2 triangles form a convex shape, flip the interior edge



A

B

heuristic: Every time I see A, change it to B.

Algorithm:

while ∃ an edge that can be flipped to be "locally Delaunay"

flip it

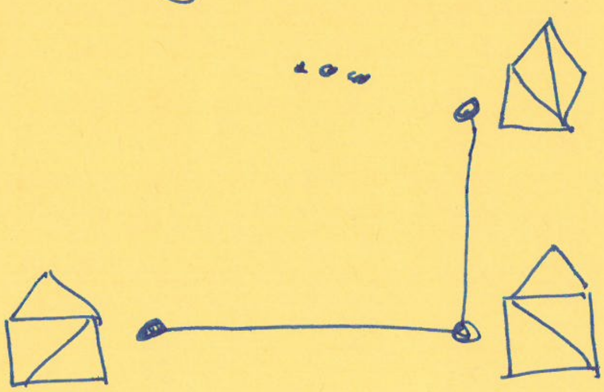
end while.

Observation: Any triangulation of n points has the same # of edges

→ NOT SO ~~too~~ obvious!
① we won't go on forever
② we won't wind up at a local minimum. ✓

Loop invariant:
i = # times loop completed,
L_i = we have at least i locally Delaunay edges

Space of all triangulations of P
- vertex = a triangulation
- add edge if exactly 1 edge flip difference



this space is (path)-connected!