Union-Find Problems

2017년 11월 8일 수요일
오후 3:36

# Union-Find Problems

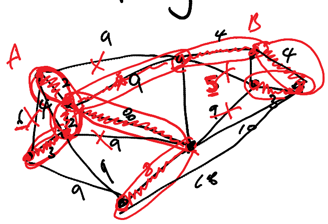
- univers : elements 1 .... n.
- initially : each element $i$ is a set by itself. (named $i$)

- Operation :
  - FIND($i$) : return the name of set containing $i$
  - UNION (A, B, C) : union elements of set A and B and call the result C (delet A & B)
  - MAKE ($x$) : make a set on element $x$

ex) Application to Minium Spanning tree (MST)

MST: Let G(V, E) be connected, undirected graph with a cost function mapped to real #. edges.

A Spanning tree is a undirected tree that connects all vertices in V. The cost of a spanning tree is just the sum of the cost of its edges. The goal is to find a spanning tree of min cost for G.

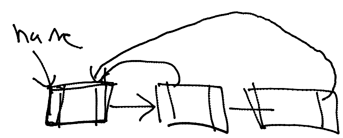



Algorithm ~~Search~~. Sketch.

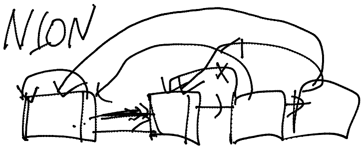
tree edge ← ∅
vertex set ← ∅
for each vertex v do
  insert {v} into vertex set
end for

while |vertex set| > 1 do
  extract edge (v, w) of lowest cost in E (edge set)
  a ← FIND(v)
  b ← FIND(w)


  if a ≠ b then
    insert (v, w) into tree edge
    UNION (a, b, a)
  endif
endwhile



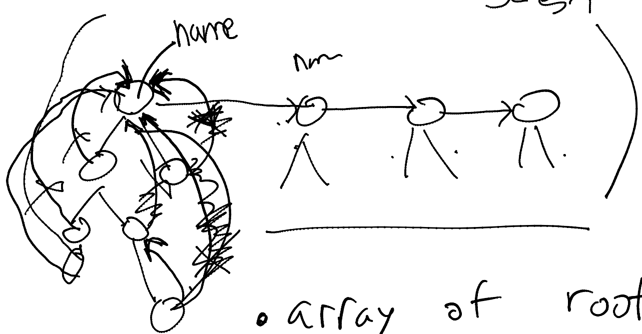

UNION



① $O(n)$

# Disjoint-set forest

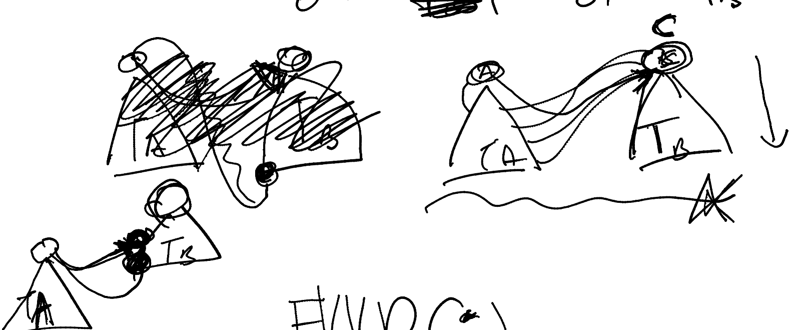
forest $\{1, 4, 7\}$




• array of roots in order to locate the root of set $i$

∘ UNION(A, B, C)

make root of $T_A$ a child of root $T_B$ and change the name of ~~root~~ parent of leaf of $T_B$ to C




$O(1)$

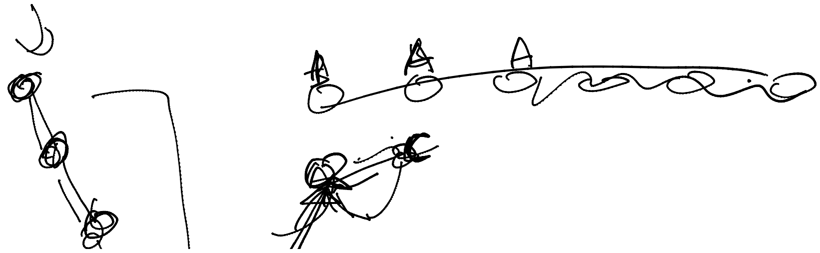
$O(n)$

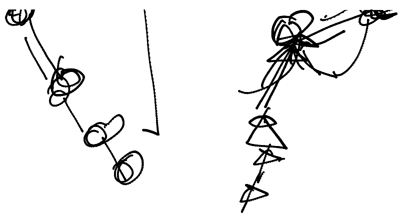
FIND(i)

following the pointer to the root.

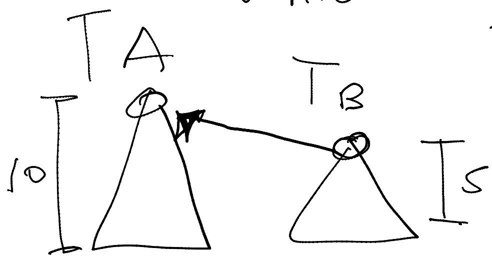
$O(n)$

Heuristics





# "Union By Rank"



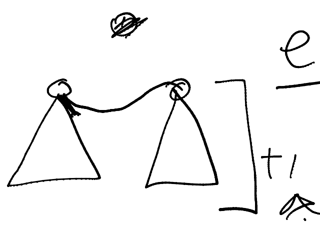

Make the root with smallest rank point to root of large rank.

LINK (x, y)

  if rank [x] > rank [y] then

    parent [y] ← x

  else parent [x] ← y




    if rank [x] = rank of [y] then

      rank [y] ← rank [y] + 1

~~else if~~ end if

No trees in the forest height greater than O(lg n)