UNION FIND

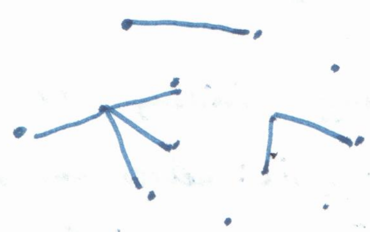
- universum $\mathcal{U} = \{1, \dots, n\}$
- rozłączne podzbiory rozbijające $\mathcal{U}$
- operacje:
  - FIND(x), $x \in \mathcal{U}$ (wynik podzbiór, do którego należy)
  - $\mathcal{U}$NION(_,_) - połączenie dwóch podzbiorów

Motywacje:

- MST (metoda Kruskala)

  n - wierzchołków m - krawędzi

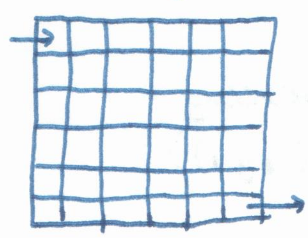
  m - # operacji FIND
  n - # operacji UNION



$e_1 \leq \dots \leq e_m$

$e_k = (v, u)$

- Konstrukcja labiryntu



na początku każde pole jest w różnej składowej, permutujemy kolejność i bierzemy po kolei pola, jak są w różnych składowych, to usuwamy ścianę, jak nie, to zostawiamy

Rozw. naiwne:

$R[0 :: |\mathcal{U}|-1]$

$R[i]$ = "nazwa" podzbioru, do którego należy

FIND: return $R[i]$ $O(1)$

UNION $O(n)$

czas wykonania σ ~ $O(n + m^2)$

Ozn: σ - ciąg m operacji FIND i n operacji UNION

2 tryby wykonania σ:
- online
- offline (tzn. możemy się najpierw przyjrzeć całej σ)

Nas dzisiaj interesuje tryb online :)

Jak to poprawić?

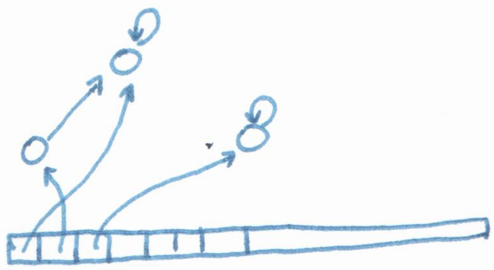
- połączyć elementy każdego podzbioru w listę
- odsyłacze do takich list
- pamiętamy rozmiary podzbiorów
- UNION: wykonujemy w sposób zbalansowany (tj. mniejszy podzbiór dołączany do większego)

Teraz wykonanie n operacji kosztuje $O(n \log n)$

Struktura drzewiasta do UF:

- każdy podzbiór - drzewo (odwrócone, tzn. każdy wierzchołek pamięta wskaźnik na ojca (nie ma wsk na syn

R

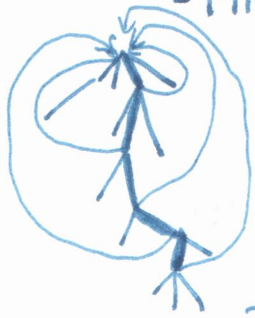


FIND(i) - przechodzimy ścieżkę do korzenia -> wynik: adres korzenia

UNINON(u,v) - połączenie korzeni drzew

2 heurystyki:

- UNION - wykonujemy w sposób zbalansowany (dzięki temu wysokości drzew są $O(\log n)$)
- FIND - z kompresją ścieżki



czyli jak raz przejdziemy raz ścieżkę, to podpinamy wszystkie wierzchołki pod korzeń.

POZOSTAJĄ DWA PYTANIA: JAK TO SZYBKO DZIAŁA? I JAK TO ZAIMPLEMENTOWAĆ?

Impl. tablicowa

R

R[i] = indeks ojca wierzchołka

Analiza kosztu wykonania σ:

Niech $\hat{\sigma}$ − σ po usunięciu operacji FIND

Def: Rząd wierzchołka v − wysokość (najdł. ścieżka do liści) w lesie powstałym po wykonaniu $\hat{\sigma}$

FAKT: Jest co najwyżej $\frac{n}{2^r}$ wierzchołków rzędu r $(\forall r)$

Uzasadnienie: Każde drzewo o wysokości h ma $\geq 2^h$ wierzchołków, drzewa wiszące pod wierzchołkiem tego samego rzędu są rozłączne

Wniosek: Maksymalny rząd $\leq \log n$

FAKT: Jeśli w trakcie wykonywania σ wierzchołek v staje się potomkiem u, to rząd(v) jest mniejszy od rzędu(u)

DEF (funkcji F)

$$F(0)=1$$
$$F(i) = 2^{F(i-1)} \quad \text{dla } i>0$$

i	$F(i)$
0	1
1	2
2	$2^2=4$
3	$2^4=16$
4	$2^{16}=65536$
5	$2^{2^{16}}=2^{65536}$

DEF ($\log^*$)

$$\log^* n = \min\{k : F(k) \geq n\}$$

DEF (grupa rzędu)

rząd g jest w grupie $\log^* g$

	n	$\log^* n$	
	0	0	grupa 0
	1	0	
	2	1	grupa 1
	3	2	grupa 2
	4	2	
	5	3	grupa 3
	6	3	
	⋮	⋮	
$F(3)$	16	3	
	17	4	grupa 4
	⋮	⋮	
$F(4)=$	65536	4	

w tej grupie jest $F(4)-F(3)$ rzędów

Koszt σ:

$\sigma: U_1, U_2, F_1, F_2, \ldots$

$$\text{Koszt}(\sigma) = \underbrace{\sum_{i=1}^{n} \text{koszt}(U_i)}_{O(n)} + \sum_{i=1}^{m} \text{koszt}(F_i)$$

Koszt F_i:

$= O(\text{długość ścieżki, którą przejdziemy})$

tym kosztem obciążymy: niektóre wierzchołki ze ścieżki i samą instr. FIND

przechodząc wierzchołek v na ścieżce za odwiedzenie go obciążymy instrukcję F_i jeśli:

- v jest korzeniem
- v jest synem korzenia
- v ma rząd w innej grupie niż jego ojciec

W pozostałych przypadkach za odwiedzenie v obciążamy v.

Widać, że Koszt(F) = suma obciążeń inst + suma obciążeń wierzchołków

Suma obciążeń pojedynczej instrukcji F:

$2 + \log^* n - 2 = \log^* n$

max rząd $\leq \log n$
max grupa $\leq \log^*(\log n) = (\log^* n) - 1$
zmian grup $\leq \log^* - 2$

Czyli m instrukcji FIND, zostanie obciążonych $\leq m \log^* n$

Suma obciążeń wierzchołków: (*)

Spostrzeżenie:
W grupie g jest $F(g) - F(g-1)$ rzędów

$(*) = \sum_{g=0}^{\log^* n}$ sumaryczne obciążenie wierzchołków o rzędach z grupy g

dla ustalonego g
jeśli v ma rząd w tej grupie, to może zostać obciążony $\leq F(g) - F(g-1)$ razy

Ile jest wierzchołków o rzędach z grupy g?

$$\sum_{i=F(g-1)+1}^{F(g)} \frac{n}{2^i} \leq \frac{n}{2^{F(g-1)+1}}\left(1+\frac{1}{2}+\ldots\right) = \frac{n}{2^{F(g-1)}} = \frac{n}{F(g)}$$

, skąd obciążenie wszystkich wierzch. z grup. g $\leq \frac{n}{F(g)}(F(g)-F(g-1)) \leq n$. Ponieważ grup jest $\leq \log^* n$, więc $(*) \leq n \log^* n$