

UNION FIND

10.05.2018r.

- universe $U = \{1, \dots, n\}$
- różne podzbiory rozbijające U
- operacje:
 - $FIND(x)$, $x \in U$ (wynik podzbioru, do którego należy)
 - $UNION(-, -)$ - połączenie dwóch podzbiorów

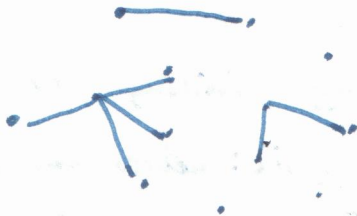
Motywacje:

- MST (metoda Kruskala)

n - wierzchołków

m - krawędzi

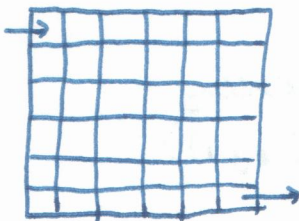
m - # operacji $FIND$
 n - # operacji $UNION$



$e_1 \leq \dots \leq e_m$

$e_k = (v, u)$

- Konstrukcja labiryntu



na początku każde pole jest w różnej składowej,
permutowujemy kolejność i bierzemy po kolei pola,
jeśli są w różnych składowych, to łączymy je
inaczej, to zostawiamy

Rów. rekurencyjne:

$R[0..|U|-1]$



$R[i]$ = "nowa" podzbiór, do którego należy

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

$UNION$ $O(n)$

czas wykonania $\sim O(n + m^2)$

Q2n: σ - ciąg operacji $FIND$ i n operacji $UNION$

2 tryby wykonania σ :

- online

- offline (tzn. możemy się najpierw popatrzeć na cały σ)

Nas dzisiaj interesuje tryb online :)

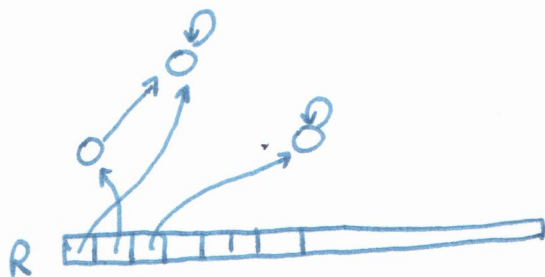
Jak to poprawić?

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

Ten ^{wykonanie} n operacji kosztuje $O(n \log n)$

Struktura danych do UF:

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



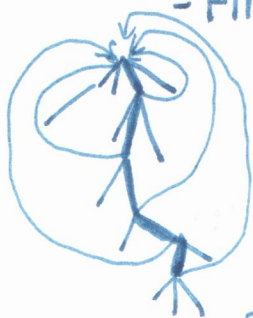
FIND(i) - przechodzący ścieżkę do korzenia \rightarrow wynik: adres korzenia

UNION(u, v) - połączenie korzeni drzew

2 heurystyki:

- UNION - wykonywany w sposób zbalansowany (drzewo lewo wysłuchuje drzewa sy $O(\log n)$)

- FIND - \approx kompresja ścieżki



czyli jak nie przejdziemy nie ścieżkę, to podpinamy wszystkie wierzchołki pod koniec.

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

Impl. tablicowe



$R[i] = \text{indeks ojca wierzchołka}$

Analiza kosztu wykonania σ :

Niech $\hat{\sigma}$ - σ po wykonaniu operacji FIND

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

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

Uzasadnienie: Każde drzewo o wysokości h ma $\geq 2^{h-1}$ wiendolków
drzewa wiszące pod wiendolkiem tego samego rzędu są rozłączne

Wniosek: Maksymalny rząd $\leq \log n$

FAKT: Jeśli u jest rodzicem wiendolka v staże 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	$2^{65536} = 2^{65536}$

DEF (\log^*)

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

DEF (grupa rzędu)

rząd g jest w grupie $\log^* g$

Koszt σ :

$$\sigma : u_1, u_2, F_1, F_2, \dots$$

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

Koszt F_i :

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

n	$\log^* n$	
0	0	} grupa 0
1	0	
2	1	} grupa 1
3	1	
4	2	} grupa 2
5	2	
6	3	} grupa 3
7	3	
8	3	
9	3	
10	3	} grupa 4
11	3	
12	3	
13	3	
14	4	}
15	4	
16	4	}
17	4	
18	4	}
19	4	
20	4	}
21	4	
22	4	}
23	4	
24	4	}
25	4	
26	4	}
27	4	
28	4	}
29	4	
30	4	}
31	4	
32	4	}
33	4	
34	4	}
35	4	
36	4	}
37	4	
38	4	}
39	4	
40	4	}
41	4	
42	4	}
43	4	
44	4	}
45	4	
46	4	}
47	4	
48	4	}
49	4	
50	4	}
51	4	
52	4	}
53	4	
54	4	}
55	4	
56	4	}
57	4	
58	4	}
59	4	
60	4	}
61	4	
62	4	}
63	4	
64	4	}
65	4	
66	4	}
67	4	
68	4	}
69	4	
70	4	}
71	4	
72	4	}
73	4	
74	4	}
75	4	
76	4	}
77	4	
78	4	}
79	4	
80	4	}
81	4	
82	4	}
83	4	
84	4	}
85	4	
86	4	}
87	4	
88	4	}
89	4	
90	4	}
91	4	
92	4	}
93	4	
94	4	}
95	4	
96	4	}
97	4	
98	4	}
99	4	
100	4	}
101	4	
102	4	}
103	4	
104	4	}
105	4	
106	4	}
107	4	
108	4	}
109	4	
110	4	}
111	4	
112	4	}
113	4	
114	4	}
115	4	
116	4	}
117	4	
118	4	}
119	4	
120	4	}
121	4	
122	4	}
123	4	
124	4	}
125	4	
126	4	}
127	4	
128	4	}
129	4	
130	4	}
131	4	
132	4	}
133	4	
134	4	}
135	4	
136	4	}
137	4	
138	4	}
139	4	
140	4	}
141	4	
142	4	}
143	4	
144	4	}
145	4	
146	4	}
147	4	
148	4	}
149	4	
150	4	}
151	4	
152	4	}
153	4	
154	4	}
155	4	
156	4	}
157	4	
158	4	}
159	4	
160	4	}
161	4	
162	4	}
163	4	
164	4	}
165	4	
166	4	}
167	4	
168	4	}
169	4	
170	4	}
171	4	
172	4	}
173	4	
174	4	}
175	4	
176	4	}
177	4	
178	4	}
179	4	
180	4	}
181	4	
182	4	}
183	4	
184	4	}
185	4	
186	4	}
187	4	
188	4	}
189	4	
190	4	}
191	4	
192	4	}
193	4	
194	4	}
195	4	
196	4	}
197	4	
198	4	}
199	4	
200	4	}
201	4	
202	4	}
203	4	
204	4	}
205	4	
206	4	}
207	4	
208	4	}
209	4	
210	4	}
211	4	
212	4	}
213	4	
214	4	}
215	4	
216	4	}
217	4	
218	4	}
219	4	
220	4	}
221	4	
222	4	}
223	4	
224	4	}
225	4	
226	4	}
227	4	
228	4	}
229	4	
230	4	}
231	4	
232	4	}
233	4	
234	4	}
235	4	
236	4	}
237	4	
238	4	}
239	4	
240	4	}
241	4	
242	4	}
243	4	
244	4	}
245	4	
246	4	}
247	4	
248	4	}
249	4	
250	4	}
251	4	
252	4	}
253	4	
254	4	}
255	4	
256	4	}
257	4	
258	4	}
259	4	
260	4	}
261	4	
262	4	}
263	4	
264	4	}
265	4	
266	4	}
267	4	
268	4	}
269	4	
270	4	}
271	4	
272	4	}
273	4	
274	4	}
275	4	
276	4	}
277	4	
278	4	}
279	4	
280	4	}
281	4	
282	4	}
283	4	
284	4	}
285	4	
286	4	}
287	4	
288	4	}
289	4	
290	4	}
291	4	
292	4	}
293	4	
294	4	}
295	4	
296	4	}
297	4	
298	4	}
299	4	
300	4	}
301	4	
302	4	}
303	4	
304	4	}
305	4	
306	4	}
307	4	
308	4	}
309	4	
310	4	}
311	4	
312	4	}
313	4	
314	4	}
315	4	
316	4	}
317	4	
318	4	}
319	4	
320	4	}
321	4	
322	4	}
323	4	
324	4	}
325	4	
326	4	}
327	4	
328	4	}
329	4	
330	4	}
331	4	
332	4	}
333	4	
334	4	}
335	4	
336	4	}
337	4	
338	4	}
339	4	
340	4	}
341	4	
342	4	}
343	4	
344	4	}
345	4	
346	4	}
347	4	
348	4	}
349	4	
350	4	}
351	4	
352	4	}
353	4	
354	4	}
355	4	
356	4	}
357	4	
358	4	}
359	4	
360	4	}
361	4	
362	4	}
363	4	
364	4	}
365	4	
366	4	}
367	4	
368	4	}
369	4	
370	4	}
371	4	
372	4	}
373	4	
374	4	}
375	4	
376	4	}
377	4	
378	4	}
379	4	
380	4	}
381	4	
382	4	}
383	4	
384	4	}
385	4	
386	4	}
387	4	
388	4	}
389	4	
390	4	}
391	4	
392	4	}
393	4	
394	4	}
395	4	
396	4	}
397	4	
398	4	}
399	4	
400	4	}
401	4	
402	4	}
403	4	
404	4	}
405	4	
406	4	}
407	4	
408	4	}
409	4	
410	4	}
411	4	
412	4	}
413	4	
414	4	}
415	4	
416	4	}
417	4	
418	4	}
419	4	
420	4	}
421	4	
422	4	}
423	4	
424	4	}
425	4	
426	4	}
427	4	
428	4	}
429	4	
430	4	}
431	4	
432	4	}
433	4	
434	4	}
435	4	
436	4	}
437	4	
438	4	}
439	4	
440	4	}
441	4	
442	4	}
443	4	
444	4	}
445	4	
446	4	}
447	4	
448	4	}
449	4	
450	4	}
451	4	
452	4	}
453	4	
454	4	}
455	4	
456	4	}
457	4	
458	4	}
459	4	
460	4	}
461	4	
462	4	}
463	4	
464	4	}
465	4	
466	4	}
467	4	
468	4	}
469	4	
470	4	}
471	4	
472	4	}
473	4	
474	4	}
475	4	
476	4	}
477	4	
478	4	}
479	4	
480	4	}
481	4	
482	4	}
483	4	
484	4	}
485	4	
486	4	}
487	4	
488	4	}
489	4	
490	4	}
491	4	
492	4	}
493	4	
494	4	}
495	4	
496	4	}
497	4	
498	4	}
499	4	
500	4	}
501	4	
502	4	}
503	4	
504	4	}
505	4	
506	4	}
507	4	
508	4	}
509	4	
510	4	}
511	4	
512	4	}
513	4	
514	4	}
515	4	
516	4	}
517	4	
518	4	}
519	4	
520	4	}
521	4	
522	4	}
523	4	
524	4	}
525	4	
526	4	}
527	4	
528	4	}
529	4	
530	4	}
531	4	
532	4	}
533	4	
534	4	}
535	4	
536	4	}
537	4	
538	4	}
539	4	
540	4	}
541	4	
542	4	}
543	4	
544	4	}
545	4	
546	4	}
547	4	
548	4	}
549	4	
550	4	}
551	4	
552	4	}
553	4	
554	4	}
555	4	
556	4	}
557	4	
558	4	}
559	4	
560	4	}
561	4	
562	4	}
563	4	
564	4	}
565	4	
56		

tym kontekstem obaczymy: niektóre ~~strony~~ wiadomości ze strony i
samym instr. FIND

przechodząc wiadomości w nie ściśle ~~dotyczące~~ go ~~dotyczą~~
za odwołanie go do przynajmniej instrukcji Fi, jeśli:

- v jest koneniem
- v jest synem koneniz
- v ma nogę w innej grze niż jego gręci

U porostajících pnyprodleech ze dmedhenie v dopriany

Wieder, $\text{Kosten}(F) = \text{summe der Ausgaben ist} + \text{summe der Ausgaben welche}$
 kon

Suma obciążeń pojedynczej instrukcji F:

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

$$\max \text{ nge} \leq \log n$$

max grade $\leq \log^*(\log n) = (\log^* n)$

زمان $g_p \leq \log^* - 2$

Cyfli m instruliji FIND zoshene
decigriovych $\leq m \log^* n$

Suma dengan penchoikan : (*)

Spostn exerie:

W grupie g jest $F(g) - F(g-1)$ nodów

(*) $\sum_{q=0}^{\log_2 n}$ сумиране доприносите от всички z групи g

dla ustalonego g

jeśli w me ngd w tej grupie, to może raczej obciążony

$$\ll F(g) - F(g-1) \text{ very}$$

He jert hierhoër o nedachs 2 grupp g?

$$\sum_{i=F(g)}^{F(g)} \frac{F(g)}{2^i} \leq 2^{\frac{n}{F(g-1)+1}} \left(1 + \frac{1}{2} + \dots\right) = \frac{2^{\frac{n}{F(g-1)+1}}}{2^{\frac{n}{F(g-1)+1}}} = \frac{n}{F(g)}, \text{ stąd doczy}$$

serie wykładnicza. Wierch. 2. grup. $g \leq \frac{n}{F(g)} (F(g) + f(g-1)) \leq \frac{n}{F(g)}$. Ponieważ grup jest $\leq \log^* n$, więc (*) $\leq \log^*$