Dawid Pawliczek

Union–Find

Dowód, że koszt zamortyzowany wynosi $O\big(n \log^* n\big)$

Definicje

- Każdy węzeł x ma atrybut $rank\,(x)$.
- `makeSet`(x) ustawia $rank(x) = 0$ i $parent(x) = x$.
- `union`(x, y) łączy dwa zbiory: korzeniem zostaje ten, którego ranga jest większa; przy rangach równych wybieramy dowolny i *zwiększamy* jego rangę o 1.
- `find`(x) zwraca korzeń zbioru oraz ustawia dla każdego odwiedzonego węzła $parent =$ korzeń (kompresja ścieżek).

Lemat 1

Twierdzenie. *Wysokość każdego drzewa jest nie większa niż ranga jego korzenia.*

Dowód. Rangę zwiększamy tylko wtedy, gdy łączymy *dwa* drzewa o tej samej randze r, wybierając jeden korzeń i nadając mu rangę $r + 1$. Wysokość nowego drzewa rośnie wówczas najwyżej o 1, czyli również maksymalnie do $r+1$. Kroki indukcyjne zachowują tezę, a dla rangi 0 jest ona oczywista. □

Uwaga: kompresja ścieżek może zmniejszyć wysokość drzewa, dlatego nierówność nie działa w drugą stronę.

Lemat 2

Twierdzenie. *Dla każdego węzła x $\;rank(x) \le \lfloor \log_2 n \rfloor$.*

Dowód. Aby ranga węzła osiągnęła wartość r, musiało się połączyć co najmniej 2^r elementów: każdy wzrost rangi wymaga scalenia dwóch drzew o tej samej randze. Zatem $2^r \le n$, skąd $r \le \log_2 n$. □

Grupowanie rang a $\log^*$

Zamiast śledzić każdy *pojedynczy* wzrost rangi (jest ich $\le \log_2 n$), grupujemy je w **poziomy**. Definiujemy

$$F(0) = 1, \quad F(i) = 2^{F(i-1)} \ (i \ge 1).$$

Poziom i zawiera wszystkie rangi $F(i-1) < r \le F(i)$. Liczba poziomów potrzebnych, by objąć maksymalną rangę $\le \log_2 n$, wynosi $\log^* n$, ponieważ $F\big(\log^* n\big) \ge n$.

Ile razy węzeł może zmienić rodzica?

Podczas `find` każdy odwiedzony węzeł v ustawia $parent(v) \leftarrow$ korzeń czyli na węzeł o *ściśle większej* randze. Węzeł może więc „awansować” co najwyżej: raz z poziomu 0 do 1, raz z 1 do 2, … Łącznie najwyżej $\log^* n$ razy.

Koszt operacji

- `union` – koszt $O(1)$.
- `find` – dla każdego odwiedzonego węzła:
  - odczyt `parent` – $O(1)$,
  - zapis nowego `parent` – *co najwyżej* $\log^* n$ razy w całej sekwencji operacji.

Przy n elementach łączny koszt wszystkich zmian rodzica to $O\big(n \log^* n\big)$, a pozostałe odczyty i `union`-y mieszczą się w $O(n)$. Stąd amortyzowany koszt sekwencji $m = O(n)$ operacji wynosi

$$O\big(n \log^* n\big).$$