Zadanie 4 lista 7

26 czerwca 2020

Spis treści

1 Rozwiązanie

Będziemy utrzymywać następujący niezmiennik:

- W korzeniu każdego drzewa znajduje się jeden kredyt
- Wierzchołek wewnętrzny ma przynajmniej 2 kredyty, jeśli utracił syna.

1.1 Insert

Operacja insert polega na utworzeniu nowego drzewa z pojedynczą wartością i dodanie go na koniec listy drzew. Operacje utworzenia i dodania na koniec listy wymagają jednego kredytu. Żeby utrzymać niezmiennik musimy dodatkowo zostawić jeden kredyt w korzeniu dodanego drzewa. Zatem operacja insert wymaga 2 kredytów.

1.2 Meld

Operacja meld polega na połączeniu dwóch list zawierających drzewa obu kopców i wybraniu wskaźnika odpowiadającego mniejszemu z minimów w obu kopcach. Potrzebujemy na to 1 kredyt.

1.3 Findmin

Operacja findmin polega na odczytaniu wartości najmniejszego elementu w kopcu, co możemy wykonać patrząc na wskaźnik MIN w kopcu. Wymaga to 1 kredytu.

1.4 Decreasekey

Operacji decreasekey będziemy przydzielać 4 jednostki kredytowe. Jeśli operacja zmniejszenia wartości klucza nie zaburzy porządku kopcowego to potrzebujemy tylko 1 kredytu na wykonanie operacji zmniejszenia wartości. Jeśli natomiast zaburza porządek kopcowy oraz ojciec tego wierzchołka nie stracił jeszcze 2 synów to odcinamy poddrzewo o korzeniu w tym wierzchołku, dodajemy je na koniec listy drzew i zostawiamy kredyt w korzeniu dodanego drzewa. Wymaga to 2 kredytów. Pozostałe 2 kredyty zostawiamy w ojcu odciętego wierzchołka. Pozostaje rozważyć przypadek, gdy decreasekey zaburza porządek kopcowy oraz ojciec tego wierzchołka stracił już dwóch synów. Odcinamy wówczas poddrzewo o korzeniu w ojcu o ile dziadek nie stracił jeszcze 2 synów (jeśli stracił to postępujemy analogicznie do bieżącego przypadku schodząc w stronę korzenia). Zauważmy, że skoro ojciec stracił już 2 synów, to oba odcięcia odłożyły w nim kredyty. Można wykorzystać je do odcięcia poddrzewa, dodania go do listy oraz odłożenia kredytu w korzeniu nowego drzewa - 2 kredyty. Pozostałe kredyty odkładamy w dziadku. Zatem na operacje decreasekey wystarczą 4 kredyty.

1.5 Deletemin

Operacja deletemin polega na usunięciu minimalnej wartości w kopcu, dodania na koniec listy poddrzew o korzeniach w synach usuniętego wierzchołka oraz połączeniu drzew znajdujących się na liście tak, aby nie istniały dwa o tym samym rzędzie (łączymy ze sobą tylko drzewa tych samych rzędów). W celu analizy wymaganej do przydzielenia dla deletemin liczby kredytów udowodnimy najpierw następujący lemat:

Lemat 1. *Dla każdego wierzchołka v kopca Fibonacciego (w których kaskadowe wykonanie operacji cut wykonywane jest dopiero wtedy, gdy wierzchołek traci trzeciego syna) o rzędzie k, drzewo zakorzenione w v ma rozmiar wykładniczy względem k.*

Dowód. Weźmy dowolny wierzchołek v kopca. Przez $y_1, \ldots, y_k$ oznaczmy synów v uporządkowanych w kolejności ich podłączania. Rozważmy wierzchołek y_i. Skoro wierzchołki $y_1, \ldots, y_{i-1}$ zostały podwieszone przed y_i to w momencie podwieszania y_i, rząd v musiał wynosić przynajmniej $i-1$. Łączymy tylko drzewa równych rzędów, stąd w momencie podwieszania rząd y_i wynosił tyle co rząd v, czyli wynosił co najmniej $i-1$. Jaki rząd może mieć teraz y_i? Mógł zmniejszyć się co najwyżej o 2, bo pozwalamy tylko na stratę 2 synów w każdym wierzchołku. Zatem obecny rząd y_i wynosi co najmniej $i-3$. Wiemy więc, że w każdym momencie dla dowolnego wierzchołka v jego i-ty syn ma co najmniej rząd $i-3$.

Niech F_k będzie najmniejszym drzewem spełniającym powyższe zależności (korzeń ma rząd k). Skoro F_k jest najmniejsze, to każdy syn korzenia jest najmniejszym możliwym poddrzewem, a skoro i-ty syn ma rząd co najmniej $i-3$ to w przypadku F_k rząd i-tego syna będzie wynosił $\max\{0, i-3\}$. Skoro poddrzewo też jest minimalne, to poddrzewo o korzeniu w i-tym synie jest drzewem $F_{\max\{0,i-3\}}$. Stąd dla drzewa F_k jego synowie są korzeniami drzew $F_0, F_0, F_0, F_1, F_2, \ldots, F_{k-4}, F_{k-3}$. Zatem liczba wierzchołków $|F_k|$ jest równa

$$1 + 2F_0 + \sum_{i=0}^{k-3} |F_i| \geqslant Fib(k-1)$$

Wiemy, że $Fib(k-1) \sim \left(\frac{1+\sqrt{5}}{2}\right)^{k-1}$, stąd liczba wierzchołków w drzewie o rzędzie k jest wykładnicza względem k. □

Z powyższego lematu wnioskujemy, że każdy wierzchołek ma stopień co najwyżej $\mathcal{O}(\log n)$.
Dla operacji deletemin należy przydzielić $\mathcal{O}(\log n)$ kredytów. Po usunięciu minimum potrzebujemy $\mathcal{O}(\log n)$ kredytów na podłączenie drzew o korzeniach w synach do listy drzew kopca oraz $\mathcal{O}(\log n)$ kredytów na odłożenie w ich korzeniach – bo rząd usuniętego wierzchołka wynosił co najwyżej $\mathcal{O}(\log n)$. Podczas przeglądania listy drzew, w celu złączenia drzew o tym samym rzędzie, koszt odwiedzenia dołączanego drzewa oraz dołączenia go do drugiego może zostać pokryty przez kredyt znajdujący się w korzeniu. Koszt odwiedzenia drzew, które nie zostaną podłączone do innego drzewa można opłacić używając $\mathcal{O}(\log n)$ kredytów – każdy wierzchołek jest stopnia co najwyżej $\mathcal{O}(\log n)$, więc mamy co najwyżej $\mathcal{O}(\log n)$ możliwych różnych stopni drzew w kopcu, a na koniec na liście nie mogą występować dwa drzewa o tym samym stopniu. Stąd ostatecznie operacja deletemin wymaga $\mathcal{O}(\log n)$ kredytów.