Algorytmy i Struktury Danych - zadanie 6.7

Maksymilian Zawartko

11 czerwca 2020

1 Zadanie

Czy można tak zmodyfikować drzewa AVL, by operacje insert, delete, search, minimum, maksimum nadal wykonywały się w czasie O(log n), a operacje następnik(v) i poprzednik(v), gdzie v jest adresem węzła, wykonywane były w czasie O(1)?

2 Idea rozwiązania

Wystarczy dodać do każdego węzła wskaźniki na jego poprzednika i następnika oraz odpowiednio zmodyfikować procedury usuwania i wstawiania elementu, by te wskaźniki aktualizowały. Wyszukiwanie nie wymaga zmiany. W sekcji "Pseudokod" podałem zmodyfikowane wersje usuwania i wstawiania, odwołujące się do standardowych procedur.

3 Pseudokod

```
def usuń(element, drzewo):
    usuń_jak_w_avl(element, drzewo)
    u <- wskaźnik na usunięty element
    if u.następnik is not None:
        u.następnik.poprzednik = u.poprzednik
    if u.poprzednik is not None:
        u.poprzednik.następnik = u.następnik
    delete(u)  # zwolnienie pamięci w miejscu, gdzie był u

def wstaw(element, drzewo):
    wstaw_jak_w_avl(element, drzewo)
    w <- wskaźnik na wstawiony element
    o <- ojciec w

    if w jest lewym synem o:
        w.następnik = o
```

```
        w.poprzednik = o.poprzednik
        o.poprzednik = w
        if w.poprzednik is not None:
            w.poprzednik.następnik = w
    else:
        w.poprzednik = o
        w.następnik = o.następnik
        o.następnik = w
        if w.następnik is not None:
            w.następnik.poprzednik = w

    wykonaj_rotacje_jak_w_avl_żeby_zrównoważyć(w, drzewo)
```

4 Uzasadnienie poprawności

4.1 Usuwanie elementu

Wskaźniki na poprzednika i następnika odpowiednio następnika i poprzednika elementu usuwanego jest ustawiany na jego odpowiednio następnika i poprzednika. Poza tym operacja usuwania wygląda dokładnie tak, jak w zwykłym drzewie AVL, a więc działa.

4.2 Wstawianie elementu

Wskaźniki na następnika i poprzednika wstawianego elementu, jak i wskaźnik na następnika poprzednika (gdy wstawiony element, przed rotacjami równoważącymi, jest lewym synem swojego ojca) lub poprzednika następnika (gdy wstawiony element jest prawym synem przed rotacjami) są odpowiednio aktualizowane. Poza tym operacja usuwania wygląda dokładnie tak, jak w zwykłym drzewie AVL, a więc działa.

5 Złożoność algorytmu

5.1 Usuwanie elementu

W oryginalnym drzewie AVL usuwanie działa w czasie O(logn), w zmodyfikowanej wersji dodana jest jedynie stała liczba operacji, więc pozostaje O(logn).

5.2 Wstawianie elementu

W oryginalnym drzewie AVL wstawianie działa w czasie O(logn), w zmodyfikowanej wersji dodana jest jedynie stała liczba operacji, więc pozostaje O(logn).