

2016 Z Z 9

Opisz (albo zapisz w pseudokodzie), w jaki sposób wykonywana jest operacja wstawiania klucza w drzewie van Emde Boasa.

```
def emptyTreeInsert(x, S):
    S.min = x
    S.max = x

def insert(x, S):
    if S.min is Nil:
        emptyTreeInsert(x, S)
    else:
        if x < S.min:
            swap(x, S.min)
        if S.cluster(H(x)) is Nil:
            insert(H(x), S.summary)
            emptyTreeInsert(L(x), S.cluster(H(x)))
        else:
            insert(L(x), S.cluster(H(x)))
        if x > S.max:
            S.max = x
```

x - klucz który chcemy dodać do struktury

S - drzewo vEB

S.min - numer najmniejszej wypełnionej pozycji w drzewie

S.summary - 'podsumowanie' drzewa S, drzewo vEB zawierające informację o S.min, S.max wszystkich poddrzew S

S.cluster - tablica poddrzew vEB

H(x) - zwraca numer grupy(poddrzewa) x

L(x) - zwraca pozycję x dla danej grupy

2017 Z Z 19

Wyjaśnij, po co pamiętane są wartości min i max w każdej strukturze rekurencyjnej w drzewach(kolejkach) van Emde Boasa

Bez spamiętywania mamy odpowiednio dla operacji:

- **successor** $T(n) = 3T(\sqrt{n}) + O(1)$
- **insert** $T(n) = 2T(\sqrt{n}) + O(1)$

Gdy spamiętujemy dla obydwu operacji mamy $T(n) = T(\sqrt{n}) + O(1)$

Co daje złożoność obydwu operacji $O(\log \log n)$

2017 P Z 19

Wyjaśnij (możliwie precyzyjnie), po co pamiętane są wartości min i max w każdej strukturze rekurencyjnej w drzewach (kolejkach) van Emde Boasa.

Bez spamiętywania

- **successor** $T(n) = 3T(\sqrt{n}) + O(1)$

operacje wywołane w najgorszym przypadku:

- wywołanie rek. dla poddrzewa w którym znajduje się nasz x
- wywołanie rek. dla summary naszego x
- wywołanie rek. dla poddrzewa z najmniejszym $y > x$

- **insert** $T(n) = 2T(\text{sqrt}(n)) + O(1)$

operacje wywołane w najgorszym przypadku:

- wywołanie rek. dla poddrzewa w którym znajduje się nasz x
- wywołanie rek. dla summary naszego x

Stąd złożoność obydwu procedur $> O(\log \log n)$

Spamiętywane

- **successor** $T(n) = T(\text{sqrt}(n)) + O(1)$

operacje wywołane w najgorszym przypadku:

- wywołanie rek. dla poddrzewa w którym znajduje się nasz x albo dla summary

- **insert** $T(n) = T(\text{sqrt}(n)) + O(1)$

operacje wywołane w najgorszym przypadku:

- wywołanie rek. dla poddrzewa w którym znajduje się nasz x albo dla summary

więc złożoność obydwu operacji $O(\log \log n)$

2018 z z 13

(a) Opisz (albo zapisz w pseudokodzie), w jaki sposób wykonywana jest operacja wstawiania kluczy w drzewach van Emde Boasa.

(b) Jaki jest koszt tej operacji?

(a)

```
def emptyTreeInsert(x, S):
    S.min = x
    S.max = x

def insert(x, S):
    if v.min is Nil:
        emptyTreeInsert(x, S)
    else:
        if x < S.min:
            swap(x, S.min)
        if S.cluster(H(x)) is Nil:
            insert(H(x), S.summary)
            emptyTreeInsert(L(x), S.cluster(H(x)))
        else:
            insert(L(x), S.cluster(H(x)))
        if x > S.max:
            S.max = x
```

x - klucz który chcemy dodać do struktury

S - drzewo vEB

S.min - numer najmniejszej wypełnionej pozycji w drzewie

S.summary - 'podsumowanie' drzewa S, drzewo vEB zawierające informację o S.min, S.max wszystkich poddrzew S

S.cluster - tablica poddrzew vEB

H(x) - zwraca numer grupy(poddrzewa) x

L(x) - zwraca pozycję x dla danej grupy

(b)

$O(\log\log(u))$

2018 P Z 5

Opisz (albo zapisz w pseudokodzie), w jaki sposób wykonywana jest operacja znajdowania minimum w drzewie van Emde Boasa. Jak kosztowna jest ta operacja?

W każdym poddrzewie struktury vEB znajduje się informacja o minimum.

```
def min(T):  
    return T.min
```

Koszt operacji to $O(1)$

2019 Z Z 7

(a) Napisać procedurę wstawiania elementu do struktury van Emde Boasa w pseudokodzie i (b) napisać jej złożoność z uzasadnieniem.

a) Pseudokod

```
def emptyTreeInsert(x, S):  
    S.min = x  
    S.max = x  
  
def insert(x, S):  
    if S.min is Nil:  
        emptyTreeInsert(x, S)  
    else:  
        if x < S.min:  
            swap(x, S.min)  
        if S.cluster(H(x)) is Nil:  
            insert(H(x), S.summary)  
            emptyTreeInsert(L(x), S.cluster(H(x)))  
        else:  
            insert(L(x), S.cluster(H(x)))  
        if x > S.max:  
            S.max = x
```

x - klucz który chcemy dodać do struktury

S - drzewo vEB

S.min - numer najmniejszej wypełnionej pozycji w drzewie

S.summary - 'podsumowanie' drzewa S, drzewo vEB zawierające informację o S.min, S.max wszystkich poddrzew S

S.cluster - tablica poddrzew vEB

H(x) - zwraca numer grupy(poddrzewa) x

L(x) - zwraca pozycję x dla danej grupy

b) Złożoność oraz uzasadnienie

$O(\log\log(u))$

2019 P Z 7

Zapisz w pseudokodzie procedurę znajdowania następnika elementu w strukturze van Emde Boasa. Napisz jaką ma ona złożoność.

