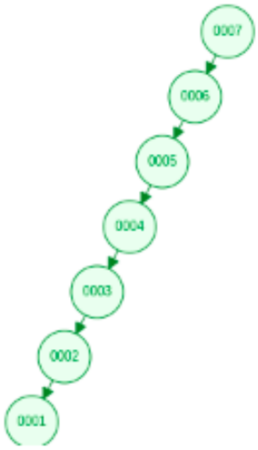
2014 P Z7

Rozważ wstawienie sekwencji kluczy do początkowo pustego drzewa splay:

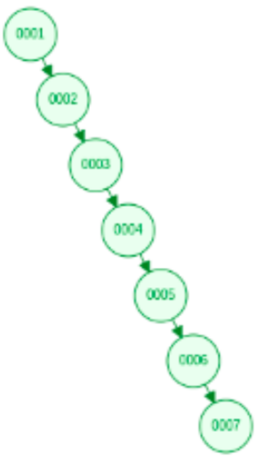
- 1, 2, 3, ..., n-1, n
- n, n-1, ..., 3, 2, 1

W którym przypadku powstanie wyższe drzewo?

Dla pierwszej sekwencji będziemy dodawali nowy wierzchołek `x` w miejsce pustego syna korzenia, a następnie rotowali w taki sposób, żeby korzeń stał się lewym synem `x`. Stan drzewa po wstawieniu kluczy 1, 2, 3, 4, 5, 6, 7



Analogicznie dla drugiej sekkwencji. Dodajemy nowy wierzchołek `x` na lewo od korzenia a następnie rotujemy go w taki sposób, że korzeń staje się prawym synem `x`. Stan drzewa po wstawieniu kluczy 7, 6, 5, 4, 3, 2, 1



Dla obydwu sekwencji otrzymamy liniowe drzewo, więc ich wysokość jest sobie równa.

2016 Z Z6

Czy trójelementowe drzewo złożone z korzenia i dwóch jego synów może być drzewem splay? Odpowiedź uzasadnij.

Drzewo splay, to BST z dodatkową operacją splay, przesuwającą serią rotacji wybrany wierzchołek do korzenia. Więc podane drzewo, może być drzewem splay. Przykład powstania takiego drzewa: insert(1), insert(3), insert(2)

2017 Z Z3

Narysuj:

(a) drzewo Splay po wykonaniu na początkowo pustym drzewie ciągu operacji:

insert(n), insert(n-1), ..., insert(2), insert(1)

(b) drzewo splay po wykonaniu operacji splay(n), splay(n-1) na drzewie otrzymanym w poprzednim punkcie

A)

https://hackmd.io/e3hdCEaNRaOFnb0HQcPrQQ

B)

https://hackmd.io/Zq2levFcQnCJGgKz9-nYNg?view

2017 P Z3

Czy po wykonaniu operacji find(x) w drzewie splay o n wierzchołkach może się zwiększyć wysokość drzewa?

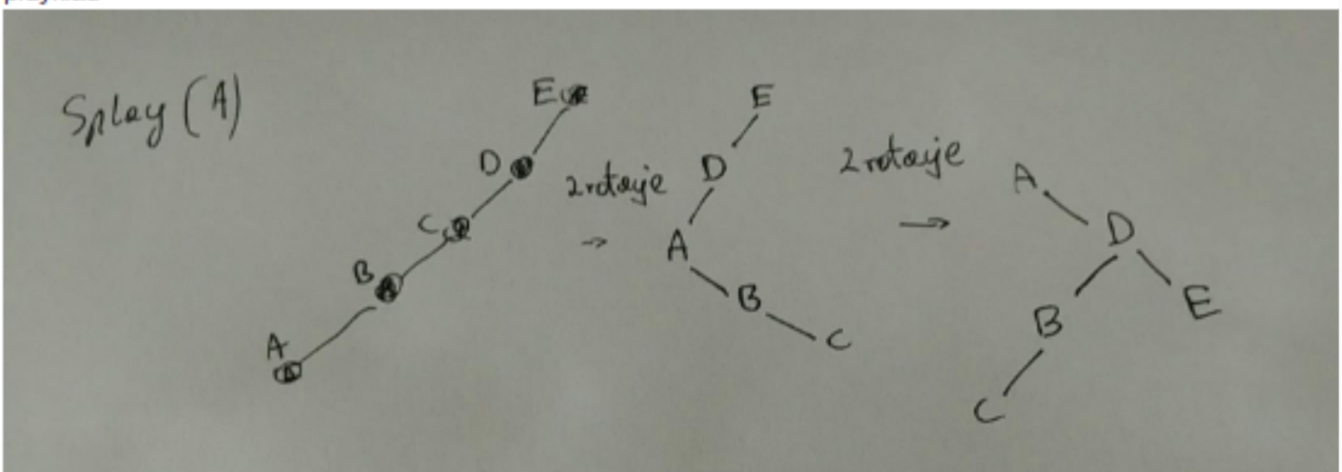
Tak, jeśli nasze drzewo jest w stanie w którym korzeń ma dwóch synów o wysokościach poddrzew `h1` i `h2`, gdzie zakładamy `h1 <= h2`, to przy pomocy operacji `find(syn z niższym poddrzewem)`, można przesunąć syna o niższym poddrzewie na miejsce roota. Wtedy `h1` zmniejszy się o 1 a `h2` wzrośnie także o 1. Stąd większe poddrzewo wzrosło i co za tym idzie wysokość drzewa także wzrosła.

2019 Z Z10

Jaka jest maksymalna liczba rotacji podczas pojedynczej operacji słownikowej na drzewach Splay o n wierzchołkach?

Mamy O(n) rotacji, ponieważ drzewo na którym wykonujemy operację może być w postaci listy.

przykład



2019 P Z10

Przedstaw w jaki sposób operacje słownikowe na drzewie BST mogą być wykonane przy pomocy stałej liczby operacji splay i stałej liczby operacji niskiego rzędu?

`find(x)` - wystarczy wykonać operację splay na `x` a potem pobrać jego wartość z korzenia

`insert(x)` - wykonujemy operację `splay(x)`, która znajduje najbliższą wartość aktualnie przetrzymywaną w słowniku dla `x` którą nazwiemy `y`. W tym momencie `y` jest rootem. Teraz mamy dwa przypadki do rozważenia:

- Jeśli `x > y` to `x.right = y.right` oraz `x.left = y`, gdzie `.right` i `.left` to odpowiednio lewy i prawy syn danego wierzchołka.

przed wykonaniem tego punktu gdzie `x=7 i y=6` **po**

- Jeśli `x < y` analogicznie do powyższego punktu

`delete(x)` - robimy `splay(x)`, zapamiętujemy `a - lewe poddrzewo x` oraz `b - prawe poddrzewo x`, a następnie usuwamy `x`. Robimy `splay(x)` na `a`, znajdujemy klucz `y`. Widzimy, że `y` jest największą wartością w `a` i nie ma mniejszej w `b`. W takim razie `y` staje się nowym korzeniem, gdzie `a` to jego lewe a `b` prawe poddrzewo