Drzewa Splay

Drzewa Splay

Są to takie drzewa BST, które ostatnio przeglądany element wynurzamy do korzenia - to sprawia, że często wyszukiwane elementy są blisko korzenia.

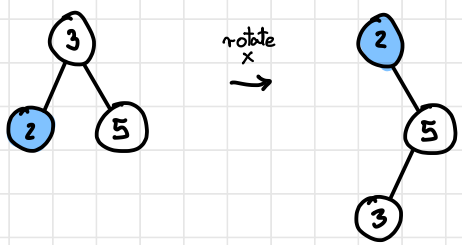
Wynurzanie jest tylko wykonywane przy wywoływaniu każdej operacji - insert, find, delete, split, join

Wynurzanie (operacja Splay)

Idąc rekurencyjnie w górę wykonujemy serię obrotów:

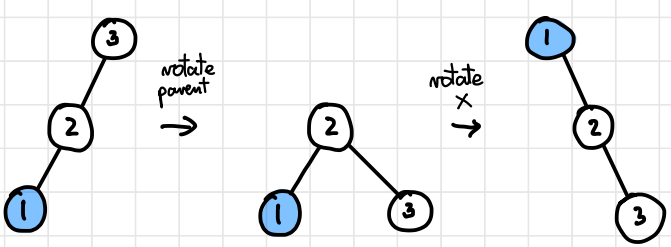
a) x ma ojca ale nie ma dziadka

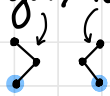
rotate (x)



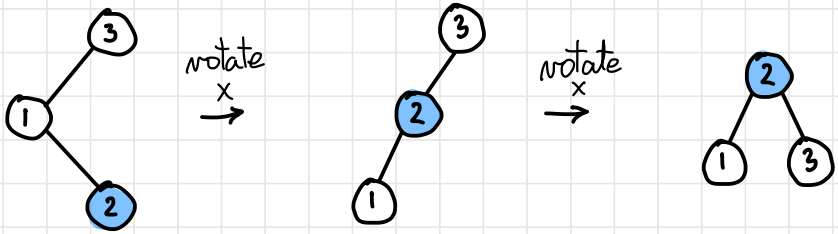
b) x ma ojca i dziadka i są razem z ojcem prawymi / lewymi dziećmi

rotate (x.parent)
rotate (x)



c) x ma ojca i dziadka. x jest prawym / lewym dzieckiem, ojciec jest lewym / prawym dzieckiem 

rotate (x)
rotate (x)



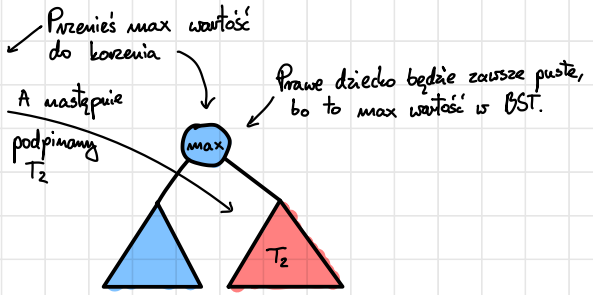
Nie trzeba się tego wszystkiego uczyć na pamięć - wystarczy intuicja, że chcemy wybrnąć węzeł utrzymując własności drzewa BST.

Jeśli elementu nie ma w drzewie to wywołujemy się na węźle najbliżej x

Join (T_1, T_2)

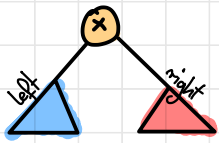
Operacja złączania dwóch splay drzew takich, że T_1 ma elementy mniejsze od T_2.

Splay(T_1, ∞)
T_1.right = T_2



Split (S, x)

```
splay (S, x)        ← Wyjmujemy x
return ( S.left, S. right)
```

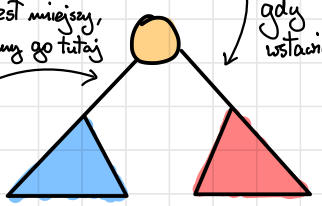


Delete (S, x)

```
l, r = split(S, x)
join (l, r)
```

Insert (S, x)

```
splay (S, x)        ← najbliższy wierzchołek do x w korzeniu
if S.root < x
   x. left = S
   x. right = S. right
   S. right = x
else
   x. right = S
   x. left = S. left
   S. left = x
```



Zauważ, że każda operacja zaczyna się od wywołania funkcji splay