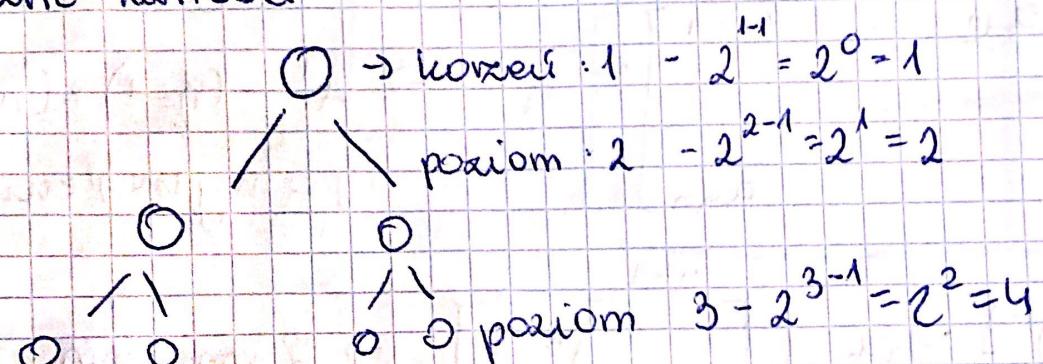


Zad 3

h - ilość poziomów, na których występuje weźmię
(Schematycznie: $h=1$; koralik i drień: $h=2 \dots$)

h - ? (ilość weźmię)

maksymalne wartości:



zatem \downarrow ilość n w zależności od poziomu h

$$\begin{matrix} h=1 \\ n=2 \end{matrix}$$

$$n \text{ poziomów} = 2^{h-1} + 2^{h-2} + 2^{h-3} \dots + 2^0$$

ciąg geometryczny $q=2$

$$Sh = \frac{1 \cdot \frac{1-2^h}{1-2}}{q} = \frac{1-2^h}{1-2} = \frac{1-2^h}{-1} = -1+2^h = 2^h - 1$$

maksymalne ilości weźmię

analgiexnie

$$n = 2^h - 1$$

• $m+1 = 2^h \Rightarrow h = \log_2(m+1)$

zadanie

find, insert, remove

node + find (node + t, int x)

h

while ($t \neq \text{null}$ $\&$ $t \rightarrow \text{key} \neq x$)

$t = (\text{left } t \rightarrow \text{key})? : t \rightarrow \text{left} : t \rightarrow \text{right}$

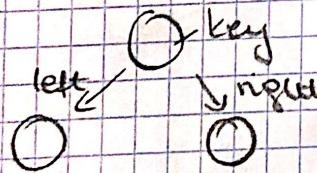
return t;

if

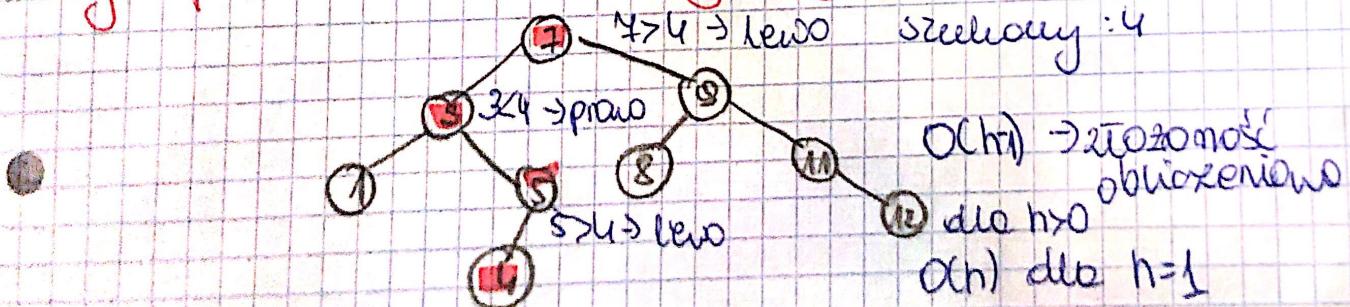
j

drzewo BST - dla każdego węzła zachodzi następujące rekurencja:

- wartości kluczy węzłów leżących pow. lewym pod drzewem węzła są mniejsze lub równe wartości klucza danego węzła
- wartości kluczy węzłów leżących w prawym poddrzewie węzła są większe lub równe wartości klucza danego węzła



Wyszukiwanie: poczynając od całego drzewa sprawdzamy kolejne kolejnych poddrzewa: jeśli którykolwiek z nich jest mniejszy od poszukiwanego to kontynuujemy w lewym poddrzewie i w przeciwnym wypadku w prawym.



```
insert (node * &t, int x)
```

```
{  
    node * t1 = &t;
```

```
    while (*t1)
```

```
{  
    t1 = x < (*t1) -> key?
```

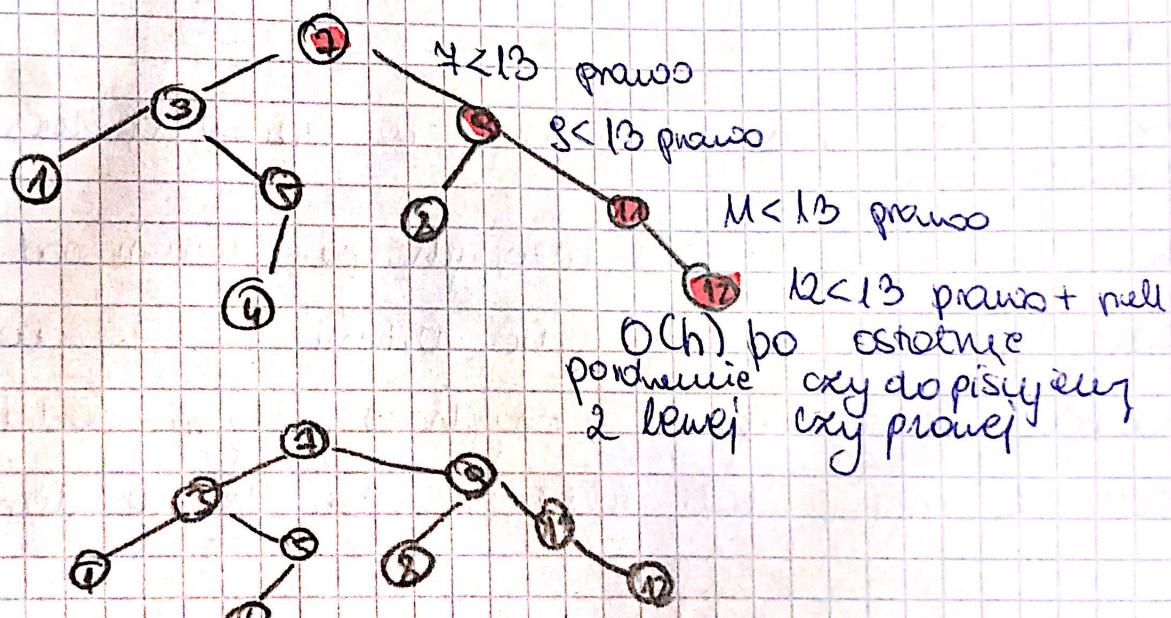
```
        &(*t1) -> left:
```

```
        &(*t1) -> right;
```

```
y
```

```
*t1 = new node (x)
```

```
insert (13)
```



remove

time $\log n$ find $O(h-1)$

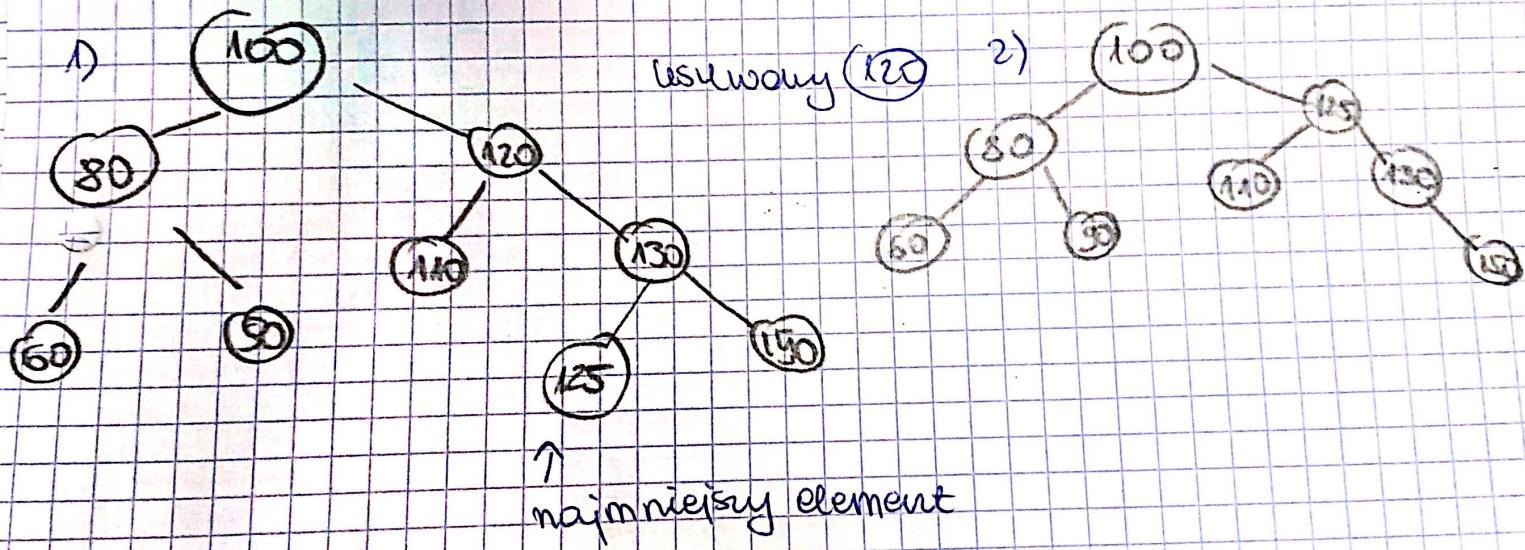
Analogiczne do algorytmu
wyszukiwania elementu należy
przeić przez dwa węzły
do mierzących, który nie
ma odpowiedniego (prawego
lub lewego) syna: dodaj tam
nowy mierzący.

Zad 12.

Procedura nie usuwa rosnącego porządku kluczy

Wpisanych w porządku inorder, dlategoże najmniejszy element w prawym poddrzewie jest zawsze mniejszy od kolejnego elementu w tym poddrzewie.

np.



Inorder:

1) 60, 80, 80, 100, 110, 120, 125, 130, 150

2) 60, 80, 80, 100, 110, 125, 130, 150