



**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE**

# **Algorytmy i struktury danych**

**Rekurencja oraz drzewa binarne**

**Piotr Kustra**

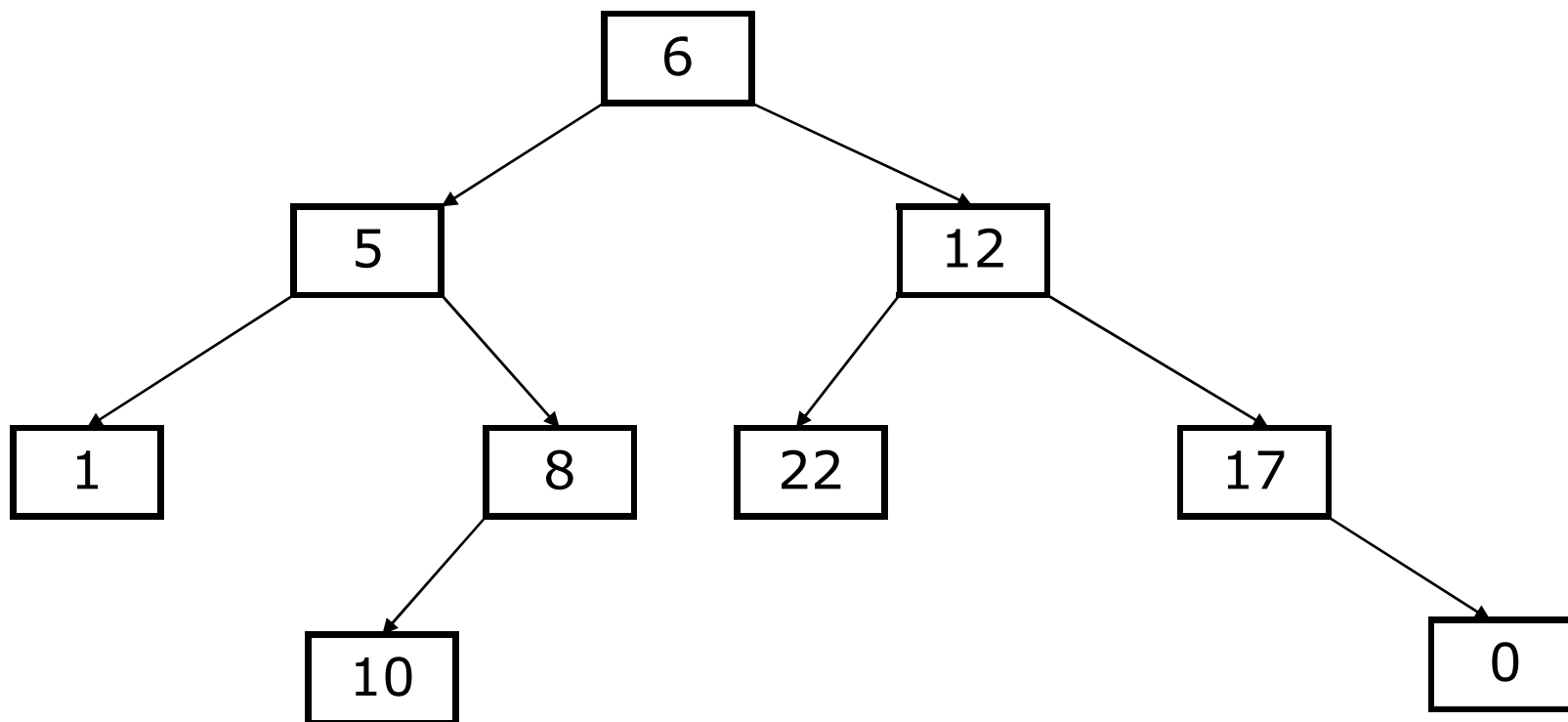
**Faculty of Metals Engineering and Industrial Computer Science  
Department of Applied Computer Science and Modeling**



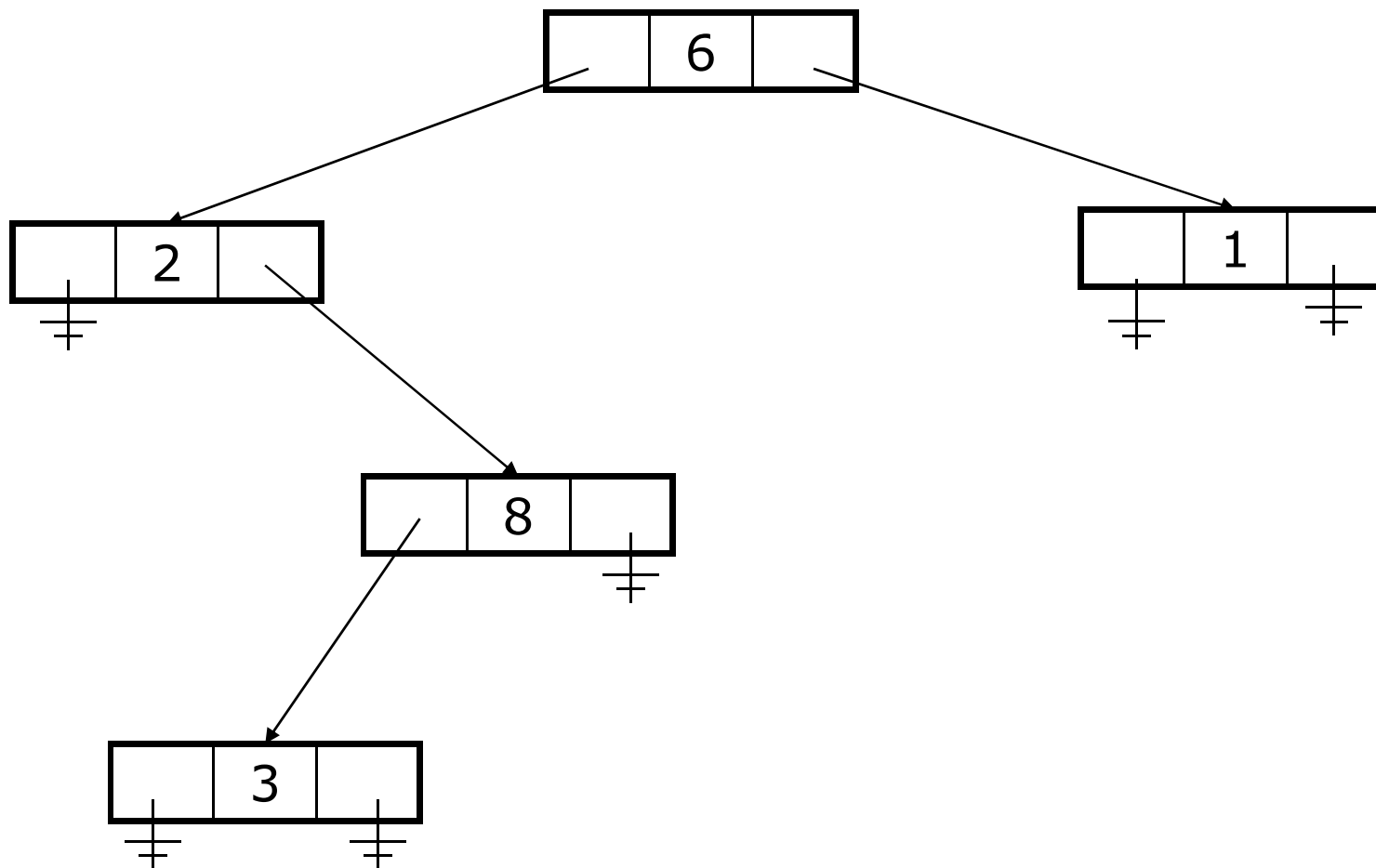
## Przykłady algorytmów rekurencyjnych

- Algorytm rekurencyjny to algorytm który wywołuje sam siebie do rozwiązania tego samego zagadnienia
- Przykłady:
  - Silnia
  - Elementy ciągu Fibonacciego
  - Sortowanie - QuickSort
  - Drzewo binarne

## Przykład drzewa binarnego

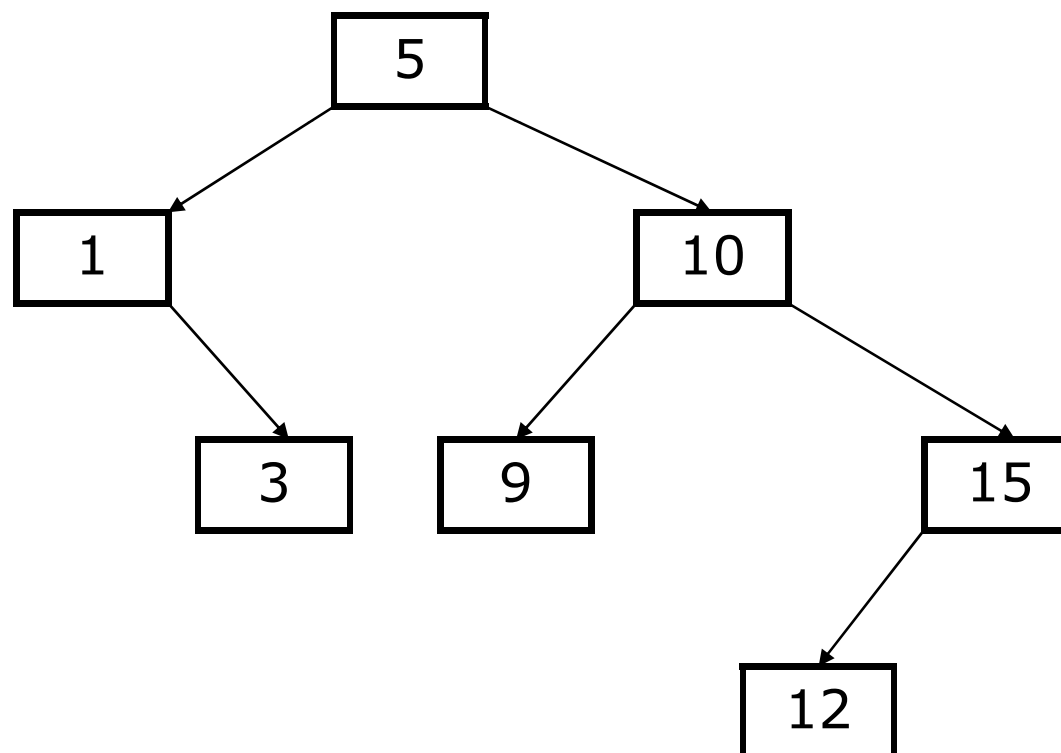


# Przykład drzewa binarnego



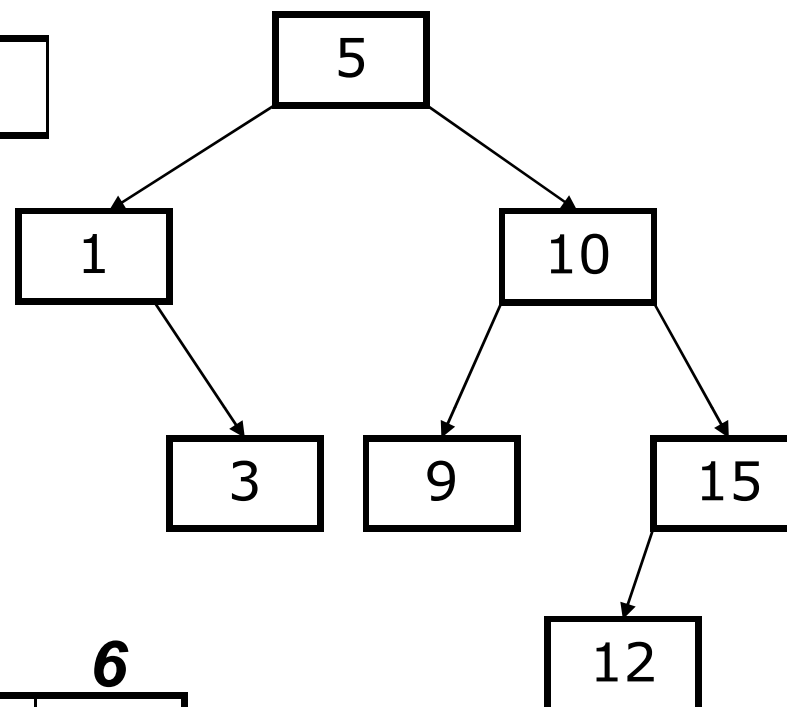
# Implementacja tablicowa

5	10	15	12	9	1	3
---	----	----	----	---	---	---



# Implementacja drzewa binarnego

5	10	15	12	9	1	3
---	----	----	----	---	---	---



	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
	5	10	15	12	9	1	3
<b>L</b>	<b>5</b>	<b>4</b>	<b>3</b>	<i>-1</i>	<i>-1</i>	<i>-1</i>	<i>-1</i>
<b>P</b>	<b>1</b>	<b>2</b>	<i>-1</i>	<i>-1</i>	<i>-1</i>	<b>6</b>	<i>-1</i>



# Implementacja drzewa binarnego

Tab – tablica „głów”, L – tablica lewych „synów”, P – tablica prawych „synów”, Licznik = 0; x – wprowadzana liczba, a - zmienna porządkowa

K1: Jeśli drzewo puste {

Wpisz wartość do tablicy

Przypisz  $L \leftarrow -1$ ,  $P \leftarrow -1$

licznik=licznik+1; }

Wpp{

Jeśli  $x > \text{tab}[a]$ {

jeśli  $P[a] > -1$ {

tab[licznik]=x

L[licznik]=P[licznik]=-1;

P[a]=licznik++;}

wpp { wykonaj krok1 dla (x i P[a]) }

}wpp

{

jeśli  $L[a] > -1$ {

tab[licznik]=x

L[licznik]=P[licznik]=-1;

L[a]=licznik++;}

wpp { wykonaj krok1 dla (x i L[a]) }

}

}



```
void wstaw(int x, int a)
{
    if(endt==0)
    {
        tab[endt]=x; l[endt]=p[endt]=-1; endt++;
    }
    else
    {
        if(x>tab[a])
        {
            if(p[a]==-1)
            {
                tab[endt]=x; l[endt]=p[endt]=-1; p[a]=endt; endt++;
            }
            else
            {
                wstaw(x,p[a]);
            }
        }
        else{
            if(l[a]==-1)
            {
                tab[endt]=x; l[endt]=p[endt]=-1; l[a]=endt; endt++;
            }
            else
            {
                wstaw(x,l[a]);
            }
        }
    }
}
```





# Sortowanie drzewa binarnego

**KR 1. Jeśli  $L[a] > -1$**

**wykonaj krok1 dla  $L[a]$**

**czytaj „głowę”**

**jeśli  $P[a] > -1$**

**wykonaj krok 1 dla  $P[a]$**



## C++

- `void sort(int a)`
- `{`
- `if(l[a]>-1)`
- `sort(l[a]);`
- `cout<<tab[a]<<" ";`
- `if(p[a]>-1)`
- `sort(p[a]);`
- `}`