

# Algorytmy i struktury danych - Struktury drzewiaste

Marcin Żurowski

16 grudnia 2019

# Struktury drzewiaste

```
/*poniższe wyjście jest tylko przykładem, program
generuje losowe drzewo*/
STRUCT-TREE T
INIT(T) /*inicjuje zmienne*/
MAKE-TREE(T,10)
PRINT-TREE(T) //drzewo losowe przykładowe drzewo
poniżej
```

# Struktury drzewiaste

```
// 1
// 3
// 2
// 6
// 7
// 4
//5
// 10
// 9
// 8
PREORDER-TREE(T) //5 6 3 1 2 7 4 9 10 8
INORDER-TREE(T) //1 3 2 6 7 4 5 10 9 8
POSTORDER-TREE(T) //1 2 3 4 7 6 10 8 9 5
CLEAR(T) /*zwalnia pamięć*/
PRINT-TREE(T) //
```

## Struktury drzewiaste - funkcje i procedury

```
STRUCT-TREE
  root = NIL
STRUCT-NODE
  key = NIL
  p = NIL
  left = NIL
  right = NIL
MAKE-TREE(T,n)
PREORDER-TREE(T)
INORDER-TREE(T)
POSTORDER-TREE(T)
PRINT-TREE(T)
```

Napisz algorytm który:

- 1 Preorder - iteracyjnie
- 2 Inorder - iteracyjnie
- 3 Postorder - iteracyjnie
- 4 Liczba węzłów w drzewie (0.5 pkt. lub zad.5)
- 5 Liczba liści w drzewie (0.5 pkt. lub zad.4)
- 6 Napisz procedurę `void make_tree_rek(Tree t, int n)`, która wygeneruje losowe drzewo w sposób rekurencyjny.  
(Uwaga: Nie używamy zmiennych statycznych, ani globalnych)  
(5 pkt. termin: do egzaminu)