

# Algorytmy i struktury danych

## Lista zadań 7 - B-drzewa

### Zadanie 1

Zapisz warunki jakie muszą spełniać klucze drzewa BST.

- Klucze w lewym poddrzewie są mniejsze od klucza w danym węźle.
- Klucze w prawym poddrzewie są większe lub równe kluczowi w danym węźle.

### Zadanie 2

Napisz procedurę `node* find(node* tree, int x)`, która zwraca wskaźnik na węzeł zawierający `x` lub `NULL`, jeśli nie ma takiego węzła.

```
node *find(node *tree, int x) // wyszukiwanie klucza (nierekurencyjne)
{
    while (tree->key != x)
        if (tree == nullptr) return nullptr;
        if (tree->key > x) tree = tree->left;
        else
            tree = tree->right;
    return tree;
}

node *find(node *tree, int x) // wyszukiwanie klucza (rekurencyjne)
{
    if (tree == nullptr) return nullptr;
    if (tree->key == x) return tree;
    if (tree->key > x) return find_recursive(tree->left, x);
    return find(tree->right, x);
}
```

### Zadanie 3

Napisz procedurę `void insert(node *tree, int x)` (dodaje do drzewa `tree` klucz `x`).

```
void insert(node *&tree, int x) // wstawianie (rekurencyjnie)
{
    if (tree == nullptr)
        tree = new node(x);

    else if (tree->key > x)
        insert(tree->left, x);
    else
        insert(tree->right, x);
}
```