

Задание 4

Деменчук Георгий ПИ19-4

Выполнить программную реализацию и сравнительный анализ быстродействия поиска элементов с помощью бинарного дерева в заданном массиве элементов: 81 77 79 68 10 12 13 20 15 24 27 42 33 51 57.

Для этого:

1. Реализовать представление данных с помощью бинарного дерева;
2. Реализовать поиск элементов в массиве с учетом времени на поиск
3. Реализовать поиск элемента в бинарном дереве с учетом времени на поиск
4. Реализовать поиск элементов в отсортированном массиве с учетом времени на поиск
5. Провести сравнительный анализ представления данных в трех программных реализациях.

Программная реализация

In [1]:

```
from sortedbinarytree_module import SortedTree
from array import array
import timeit
import random
```

In [2]:

```
def search_arr(a, search_e):
    """Реализовать поиск элементов в массиве с учетом времени на поиск"""
    for e in a:
        if search_e == e:
            print("Элемент найден")
            break
    else:
        print("Элемент не найден")

def search_tree(thistree, search_e):
    """Реализовать поиск элемента в бинарном дереве с учетом времени на поиск"""

    #print(thistree)
    #Если есть результат
    if thistree is None:
        print("Элемент не найден")
        return

    elif thistree.get_root_val() == search_e:
        print("Элемент найден")
        return

    #Если значение меньше - идем влево
    elif search_e < thistree.get_root_val():
        search_tree(thistree.get_left_child(), search_e)

    #Если значение больше или равно - идем вправо
    elif search_e >= thistree.get_root_val():
        search_tree(thistree.get_right_child(), search_e)

    #Если что-то аномальное
    else:
        raise ValueError("Что-то странное сейчас произошло")
```

In [3]:

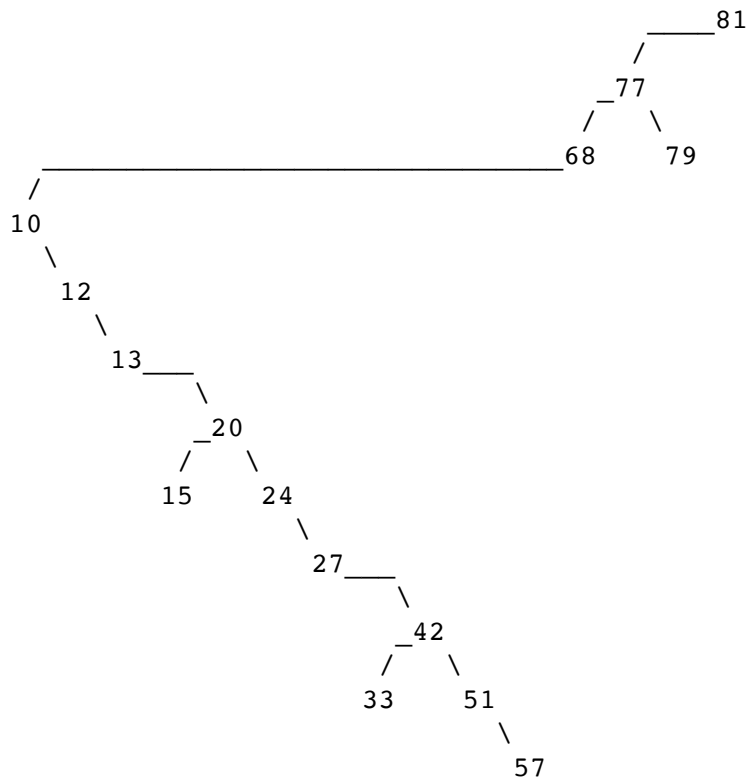
```
#Элемент для поиска
search_e = 42
items = array("i", [81, 77, 79, 68, 10, 12, 13, 20, 15, 24, 27, 42, 33, 51, 57])
```

Представление данных с помощью бинарного дерева

In [4]:

```
tree = SortedTree()
for value in items:
    tree.push(value)

print(tree)
```



Поиск элементов в массиве с учетом времени на поиск

In [15]:

```
print("Поиск элементов в массиве")
a = timeit.default_timer()
search_arr(items, search_e)
print("Время: {}".format(timeit.default_timer() - a))
```

Поиск элементов в массиве

Элемент найден

Время: 0.0003561959999984765

Поиск элемента в бинарном дереве с учетом времени на поиск

In [19]:

```
print("Поиск элементов в бинарном дереве")
a = timeit.default_timer()
search_tree(tree.root, search_e)
print("Время: {}".format(timeit.default_timer() - a))
```

Поиск элементов в бинарном дереве

Элемент найден

Время: 0.0001468999999971743

Поиск элементов в отсортированном массиве с учетом времени на поиск

In [16]:

```
print("Поиск элементов в отсортированном массиве")
items2 = sorted(list(items))
items2 = array("i", items2)
a = timeit.default_timer()
search_arr(items2, search_e)
print("Время: {}".format(timeit.default_timer() - a))
```

Поиск элементов в отсортированном массиве

Элемент найден

Время: 0.000276001999999966435

In []: