

Лабораторная работа №1:

Разработайте программу на языке Python, которая будет строить бинарное дерево (дерево, в каждом узле которого может быть только два потомка). Отображение результата в виде словаря (как базовый вариант решения задания). Далее исследовать другие структуры, в том числе доступные в модуле collections в качестве контейнеров для хранения структуры бинарного дерева. Дерево должно обладать следующими свойствами: В корне дерева (root) находится число, которое задает пользователь (индивидуально для студента). Высота дерева (height) задается пользователем (индивидуально для студента). Левый (left leaf) и правый потомок (right leaf) вычисляется с использованием алгоритмов, индивидуальных для каждого студента в группе и приведен ниже. $Root = 1$; $height = 5$, $left_leaf = root * 2$, $right_leaf = root + 3$

Рекурсивный вариант:

```
ЛР-1 > recursive.py > gen_bin_tree
1  def gen_bin_tree(root=0, height=0):
2      tree = {}
3
4      def tree_build(root2, height2):
5          if height2 > 0:
6              tree[root2] = []
7              left_leaf = root2 * 2
8              right_leaf = root2 + 3
9
10             tree[root2].append(tree_build(left_leaf, height2 - 1))
11             tree[root2].append(tree_build(right_leaf, height2 - 1))
12
13             return root2
14         if height == 0:
15             return {root: []}
16         tree_build(root, height)
17         return tree
18
19     def main():
20         root = 1
21         height = 5
22         print(gen_bin_tree(root, height))
23
24     if __name__ == "__main__":
25         main()
```

Результат:

```
PS C:\Users\sambu\OneDrive\Рабочий стол\Учеба в Герцена\II курс\Программирование Python, 2 курс, 2 семестр> & C:\Users\sambu\AppData\Local\Programs\Python\Python313\python.exe "c:\Users\sambu\OneDrive\Рабочий стол\Учеба в Герцена\II курс\Программирование Python, 2 курс, 2 семестр\ЛР-1\recursive.py"
{1: [2, 4], 2: [4, 5], 4: [8, 7], 8: [16, 11], 16: [32, 19], 11: [22, 14], 7: [14, 10], 14: [28, 17], 10: [20, 13], 5: [10, 8], 20: [40, 23], 13: [26, 16], 32: [64, 35], 19: [38, 22], 22: [44, 25], 28: [56, 31], 17: [34, 20]}
```

Тест:

```
PS C:\Users\sambu\OneDrive\Рабочий стол\Учеба в Герцена\II курс\Программи  
:/Users/sambu/OneDrive/Рабочий стол/Учеба в Герцена/II курс/Программирова  
...  
-----  
Ran 3 tests in 0.001s  
  
OK
```

Нерекурсивный вариант:

```
ЛР-1 > nonrecursive.py > ...  
1  def gen_bin_tree(root, height):  
2      tree = {}  
3  
4      if height == 0:  
5          return {root: []}  
6      stack = [(root, height)]  
7  
8      while stack:  
9          current, current_height = stack.pop()  
10  
11         if current_height > 0:  
12             tree[current] = []  
13             left_leaf = current * 2  
14             right_leaf = current + 3  
15  
16             stack.append((left_leaf, current_height - 1))  
17             stack.append((right_leaf, current_height - 1))  
18             tree[current].append(left_leaf)  
19             tree[current].append(right_leaf)  
20  
21         return tree  
22  
23     def main():  
24         root = 1  
25         height = 5  
26         print(gen_bin_tree(root, height))  
27  
28     if __name__ == "__main__":  
29         main()
```

Результат:

```
PS C:\Users\sambu\OneDrive\Рабочий стол\Учеба в Герцена\II курс\Программирование Python, 2 курс, 2 семестр> & C:\Users\sambu\AppData\Local\Programs\Python\Python313\python.exe -c
"/Users/sambu/OneDrive/Рабочий стол/Учеба в Герцена/II курс/Программирование Python, 2 курс, 2 семестр/ПР-1/nonrecursive.py"
{1: [2, 4], 4: [8, 7], 7: [14, 10], 10: [20, 13], 13: [26, 16], 20: [40, 23], 14: [28, 17], 17: [34, 20], 28: [56, 31], 8: [16, 11], 11: [22, 14], 22: [44, 25], 16: [32, 19], 19:
[38, 22], 32: [64, 35], 2: [4, 5], 5: [10, 8]}
```

Тест:

```
PS C:\Users\sambu\OneDrive\Рабочий стол\Учеба в Герцена\II курс\Программирование Python, 2 курс, 2 семестр> & C:\Users\sambu\AppData\Local\Programs\Python\Python313\python.exe -c
"/Users/sambu/OneDrive/Рабочий стол/Учеба в Герцена/II курс/Программирование Python, 2 курс, 2 семестр/ПР-1/nonrecursive.py"
.....
-----
Ran 4 tests in 0.001s

OK
```