

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

С. Ю. Гуков

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

ВЫСОТА ДЕРЕВА

по курсу:

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4326

подпись, дата

Г. С. Томчук

инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Цель работы.....	3
2	Задание.....	3
3	Краткое описание хода разработки.....	3
4	Исходный код программы	4
5	Результаты работы программы с примерами.....	5
6	Выводы	6

1 Цель работы

Целью данной лабораторной работы является изучение структуры и особенностей работы деревьев как базового элемента алгоритмов и структур данных; разработка алгоритма вычисления высоты дерева, заданного в виде массива родительских связей, с учётом производительности и возможности обработки больших деревьев (сотни тысяч вершин).

2 Задание

По заданию было необходимо:

- Разработать программу для вычисления высоты дерева, заданного в формате массива родительских связей.
- Использовать эффективный алгоритм с временной сложностью $O(n)$, чтобы обработка деревьев с $n \leq 10^5$ происходила в разумное время.
- Код программы должен быть структурирован: функции должны выполнять определённые задачи, а переменные иметь осмысленные имена.
- Продемонстрировать корректность работы программы на примерах.

3 Краткое описание хода разработки

1. Исходные данные представляют дерево через массив `parent`, где `parent[i]` указывает на родителя вершины i . Если `parent[i]=-1`, то вершина i является корнем дерева. Основной задачей является нахождение максимальной высоты, определяемой как максимальная длина пути от корня до любого листа.
2. Для достижения высокой производительности был выбран подход с «мемоизацией». Для каждого узла вычисляется высота пути до корня один раз, а затем сохраняется для дальнейшего использования.
3. Разработана функция `compute_height`, вычисляющая высоту поддерева рекурсивно. Алгоритм хранит промежуточные результаты в массиве `heights`, чтобы избежать повторных

вычислений. Основной цикл проходит по всем вершинам, чтобы найти максимальную высоту.

4. Корректность алгоритма проверена на тестовых данных из задания, а также на других примерах.

4 Исходный код программы

```
# Вычисляет высоту дерева на основе массива parent.
def calculate_tree_height(n, parent):
    # Кэш для хранения высоты поддеревьев, чтобы избежать повторных вычислений
    heights = [-1] * n

    # Рекурсивно вычисляет высоту поддерева с корнем в node.
    def compute_height(node):
        if heights[node] != -1:
            return heights[node] # Если высота уже вычислена, вернуть её

        if parent[node] == -1:
            heights[node] = 1 # Корень дерева имеет высоту 1
        else:
            heights[node] = 1 + compute_height(parent[node]) # Высота = 1 +
# высота родителя
            return heights[node]

    # Вычисляем высоту для всех узлов
    return max(compute_height(i) for i in range(n))

def main():
    # Считываем количество вершин
    n = int(input("Введите количество вершин: "))

    # Считываем массив parent
    parent = list(map(int, input("Введите массив родительских связей:
").split()))

    # Проверяем корректность ввода
    if len(parent) != n:
        print("Длина массива parent должна совпадать с n.")
        return

    # Вычисляем высоту дерева
    tree_height = calculate_tree_height(n, parent)

    # Выводим результат
    print("Высота дерева:", tree_height)

if __name__ == "__main__":
    main()
```

5 Результаты работы программы с примерами

На рис. 1, 2, 3, изображено тестирование программы с различными входными данными.

```
Введите количество вершин: 5
Введите массив родительских связей: 4 -1 4 1 1
Высота дерева: 3

Process finished with exit code 0
```

Рисунок 1

```
Введите количество вершин: 5
Введите массив родительских связей: -1 0 4 0 3
Высота дерева: 4

Process finished with exit code 0
|
```

Рисунок 2

```
Введите количество вершин: 10
Введите массив родительских связей: -1 0 0 1 1 2 2 3 .
Высота дерева: 4

Process finished with exit code 0
```

Рисунок 3

6 Выводы

- В ходе выполнения лабораторной работы была изучена структура деревьев и способы их представления в виде массива родительских связей.
- Реализован эффективный алгоритм для вычисления высоты дерева, работающий за $O(n)$.
- Получен практический опыт работы с деревьями, мемоизацией и оптимизацией алгоритмов.
- Проведена проверка программы на корректность и производительность, подтверждена её способность обрабатывать большие деревья.
- Работа помогла закрепить знания о деревьях и способах их обработки.