МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

“Московский государственный университет геодезии и картографии”

(МИИГАИК)

Факультет геоинформатики и информационной безопасности

Кафедра геоинформационных систем и технологий

**Лабораторная работа №7**

**"** **Методы решения алгоритмических задач "**

Проверил: Выполнил:

Лебедев Евгений Денисович Студент группы: 2024-ФГИИБ-ПИ-1б

Центнер Валерий Викторович

Москва 2025

**=======================================**

**Задача на деревья**

**№2 “Дерево-анаграмма”**

Ссылка на GitHub репозиторий с файлами:

<https://github.com/guguker/InfAlgoLebedev/tree/main/second_semestr/homework%207/Tree%20TASK>

Что такое дерево?

Дерево — это связный ациклический граф, является самым распространённым классом графов, применяемых в программировании.

Когда нам удобно это применять?

Помимо организации поиска, деревья используют, когда разбирают математические выражения и компьютерные программы.

Еще их используют, чтобы хранить данные для алгоритмов сжатия, а также они лежат в основе других структур данных, например, очереди с приоритетом, кучи и словари

Описание подхода к решению этой задачи:

Для решения задачи о дереве-анаграмме необходимо:

1. Создать алгоритм проверки симметричности дерева

относительно его центра.

2. Для проверки симметричности нужно сравнить левое и правое

поддерево на каждом уровне.

3. Используем рекурсивный подход для обхода дерева и сравнения

узлов.

4. Дерево является анаграммой только если значения узлов и

структура зеркально симметричны.

Условие самой задачи:

Гоша и Алла играют в игру «Удивительные деревья».

Помогите ребятам определить, является ли дерево, которое им встретилось, деревом-анаграммой?

Дерево называется анаграммой, если оно симметрично относительно своего центра.

Примеры тестовых входных и выходных данных:

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Листинг кода для решения задачи:



**=======================================**

**Динамическое программирование и жадные алгоритмы**

**№5 “Прыжки по лестнице”**

Ссылка на GitHub репозиторий с файлами:

<https://github.com/guguker/InfAlgoLebedev/tree/main/second_semestr/homework%207/Dynamic%20programming%20and%20greedy%20algorithm%20TASK>

Что такое динамическое программирование?

Что такое жадный алгоритм?

Динамическое программирование – способ разбиения задачи на подзадачи и последующее соединение результатов.

Жадный алгоритм – алгоритм, выбирающий лучшее решение “в моменте”, надеясь получить лучшее решение в целом.

Когда нам удобно это применять?

Когда есть возможность разбить задачу на подзадачи, а также важна локальная оптимизация на каждом шагу.

Описание подхода к решению этой задачи:

Для решения этой задачи будем использовать динамическое программирование:

1. Создадим массив ararray[n], где ararray[i] - количество способов

допрыгать до i-й ступеньки.

2. На каждую ступеньку можно попасть прыжком длиной от 1 до k

с предыдущих ступенек.

3. Для каждой ступеньки i суммируем количество способов

допрыгать до неё со всех доступных предыдущих ступенек.

4. Используем остаток от деления на 10⁹ + 7 для больших чисел.

Условие самой задачи:

Алла хочет доказать, что она умеет прыгать вверх по лестнице быстрее всех.

С каждой её ступеньки можно прыгнуть вверх на любое расстояние от 1 до k. Число k придумывает Алла.

Гоша не хочет проиграть, поэтому просит вас посчитать количество способов допрыгать от первой ступеньки до n-й. Изначально все стоят на первой ступеньке.

Примеры тестовых входных и выходных данных:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Листинг кода для решения задачи:

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

**=======================================**

**Решение задачи методом декомпозиции**

**№7 “Поиск пика”**

Ссылка на GitHub репозиторий с файлами:

<https://github.com/guguker/InfAlgoLebedev/tree/main/second_semestr/homework%207/Decomposion%20method%20TASK>

Что такое метод разделяй и властвуй?

Метод декомпозиции — разбиение задачи на подзадачи, решение которых объединяется для итогового результата.

Когда нам удобно это применять?

Задачи с естественной рекурсивной структурой (например, сортировка слиянием).

Описание подхода к решению этой задачи:

Для решения задачи используем метод декомпозиции, разбив задачу на следующие подзадачи:

1. Функция чтения входных данных.

2. Функция проверки, является ли элемент пиковым.

3. Функция поиска пикового элемента с использованием бинарного поиска.

4. Основная функция, объединяющая все компоненты.

Условие самой задачи:

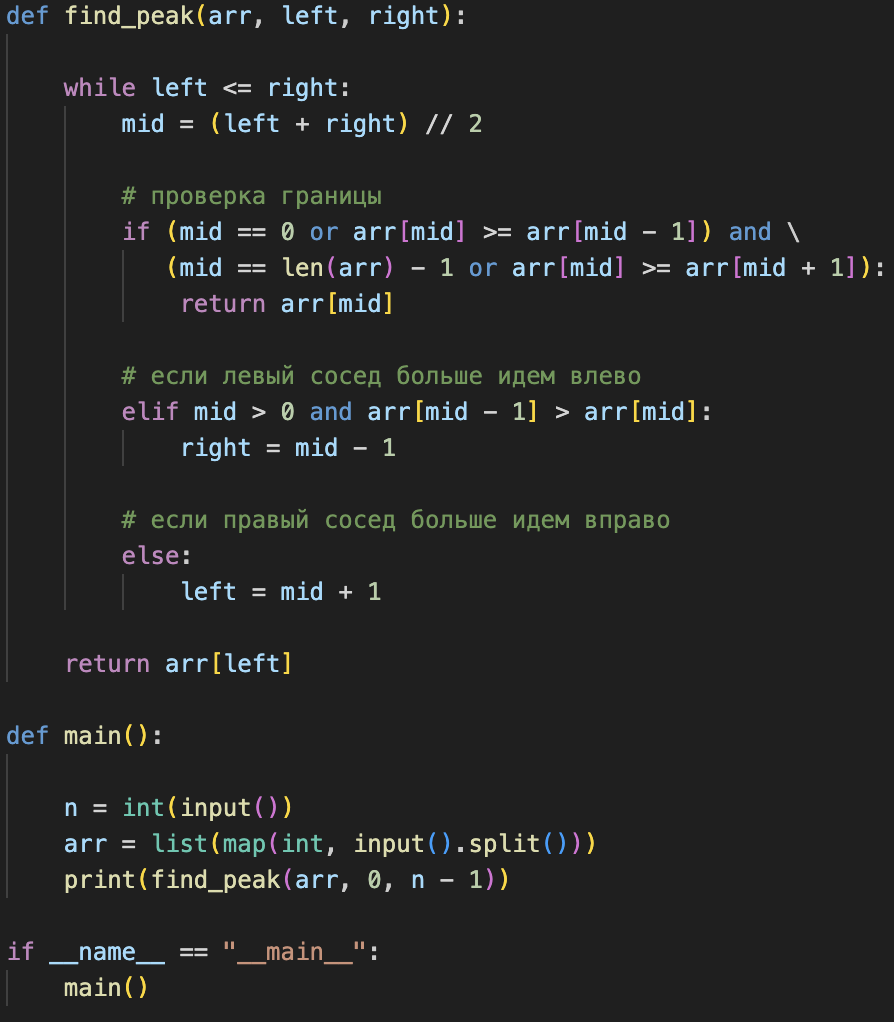
В Алгосах появился таинственный массив, в котором нужно найти пиковый элемент — такой, который не меньше своих соседей. Используйте алгоритм бинарного поиска (вариант «разделяй и властвуй») для эффективного поиска.

Примеры тестовых входных и выходных данных:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Листинг кода для решения задачи



**=======================================**

**Решение задачи методом поиск с возвратом**

**№3 “Сумма подмножества”**

Ссылка на GitHub репозиторий с файлами:

<https://github.com/guguker/InfAlgoLebedev/tree/main/second_semestr/homework%207/Search%20with%20return%20TASK>

Что такое метод поиск с возвратом?

Алгоритм поиска с возвратом тоже относится к полным переборам, но у него есть особенность, которая делает этот перебор проще:

если алгоритм понимает, что идёт по неверному пути, то все остальные варианты в этом пути тоже помечаются как неправильные и не рассматриваются.

Когда нам удобно это применять?

Поиск с возвратом чаще всего применяется в задачах комбинаторной оптимизации.

Описание подхода к решению этой задачи:

Поиск с возвратом в данной задаче реализуется следующим образом:

1. Мы последовательно перебираем все возможные комбинации

чисел.

Для каждого числа принимаем решение - берём его или нет.

2. Процесс перебора:

На каждом шаге у нас есть два варианта:

Взять текущее число в сумму,

Пропустить текущее число.

3. Условия остановки:

Успешное завершение - текущая сумма равна целевой

Неуспешное завершение:

Сумма превысила целевую,

Достигнут конец массива,

Все варианты перебраны.

Условие самой задачи:

Гоша собирает команду.

У него есть список из n кандидатов с разными баллами. Он хочет узнать, можно ли выбрать подмножество кандидатов, сумма баллов которых равна заданному числу k.

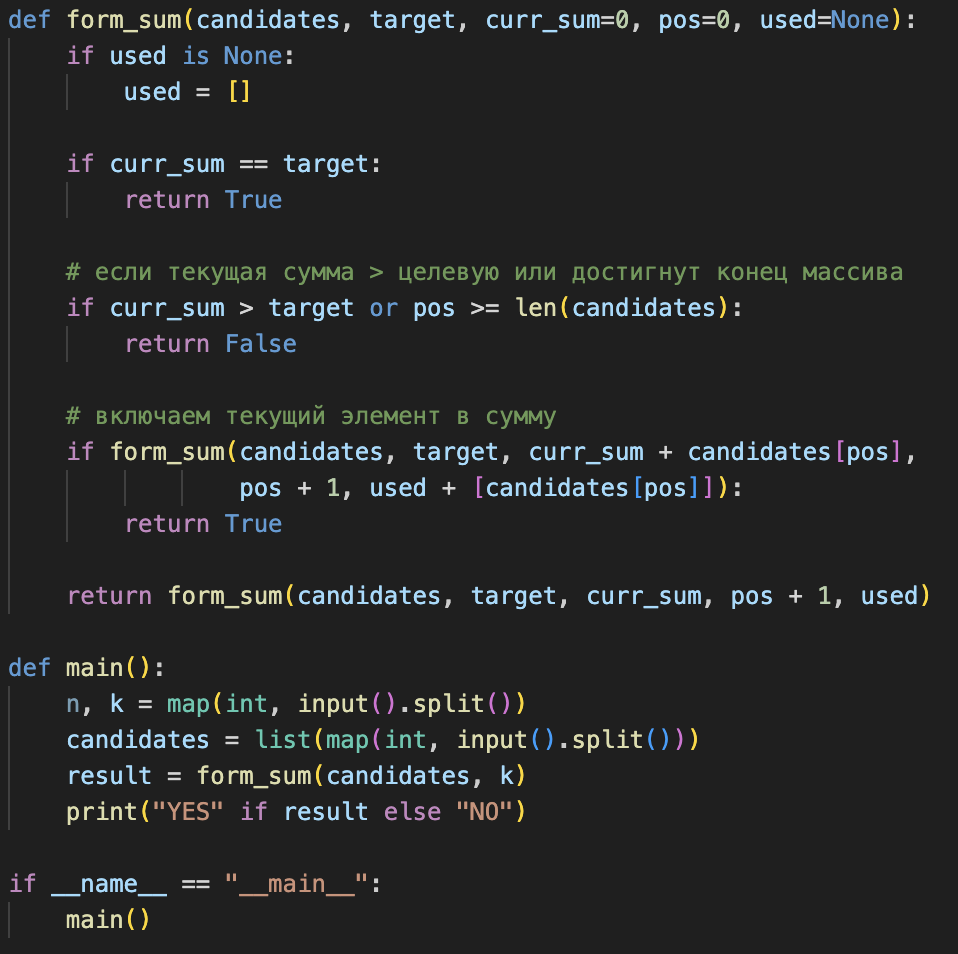
Решите задачу с помощью поиска с возвратом.

Примеры тестовых входных и выходных данных:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Листинг кода для решения задачи:



**=======================================**