|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | ИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий |
|  |
| Кафедра Математического обеспечения и стандартизации информационных технологий |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 8.2** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема:**  **Тема: «Алгоритмические стратегии или методы разработки алгоритмов. Перебор и методы его сокращения.»** | |
|  | |
| Выполнил студент группы ИНБО-12-23 | Албахтин И.В. |
| Принял ассистент | Муравьёва Е.А. |

Москва 2024

**СОДЕРЖАНИЕ**

[1. Постановка задачи 3](#_Toc184759324)

[Задание №1: 3](#_Toc184759325)

[Задание №2: 3](#_Toc184759326)

[2. Решение задачи 4](#_Toc184759327)

[Задание №1 4](#_Toc184759328)

[Задание №2 7](#_Toc184759329)

[3. Вывод 8](#_Toc184759330)

**Цель работы**: Получить навыки применения методов, позволяющих сократить число переборов в задачах, которые могут быть решены только методом перебора всех возможных вариантов решения.

# Постановка задачи

## **Задание №1:**

Ответьте на вопросы и выполните упражнения:

1. В чём суть метода грубой силы (прямолинейное решение, метод решения «в лоб»).

1. В чём суть метода «разделяй и властвуй» (декомпозиция)? Приведите примеры задач, которые решаются этим способом? Какие алгоритмы можно отнести к этому методы?
2. В чём суть метода динамического программирования? Приведите примеры задач, которые решаются этим способом?
3. В чём суть «жадного метода»? Приведите примеры задач, которые решаются этим способом?
4. В чём суть «метода ветвей и границ»? Приведите примеры задач, которые решаются этим способом?

## Задание №2:

1. Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.
2. Оценить количество переборов при решении задачи стратегией «в лоб» - грубой силы. Сравнить с числом переборов при применении метода
3. Оформить отчет в соответствии с требованиями

Таблица 1 – Вариант задания №2

|  |  |  |
| --- | --- | --- |
| **№** | **Задача** | **Метод** |
| 3 | Дана строка из заглавных букв латинского  алфавита. Найти длину наибольшего  палиндрома, который можно получить  вычеркиванием некоторых букв из данной  строки. | Динамическое  программирование |

# 2. Решение задачи

## Задание №1

**1. Метод грубой силы**

Суть метода грубой силы заключается в переборе всех возможных вариантов решения задачи и выборе наилучшего из них. Это самый простой, но часто и самый неэффективный подход, особенно для задач с большим пространством поиска. Его используют, когда другие методы слишком сложны или неизвестны, или когда пространство поиска относительно невелико.

Пример: Поиск максимального элемента в массиве методом грубой силы — это сравнение каждого элемента с текущим максимумом. Хотя для этого есть более эффективные алгоритмы, этот пример демонстрирует суть подхода.

**2. Метод «разделяй и властвуй» (декомпозиция):**

Метод «разделяй и властвуй» — это рекурсивный подход к решению задачи, который заключается в следующем:

1. Раздели: Задача разбивается на несколько подзадач меньшего размера.

2. Властвуй: Подзадачи решаются рекурсивно (или напрямую, если они достаточно малы).

3. Объедини: Решения подзадач объединяются в решение исходной задачи.

Этот метод эффективен, когда задача может быть легко разделена на более мелкие, независимые подзадачи, а решение исходной задачи легко строится из решений подзадач.

• Примеры задач:

\* Сортировка слиянием (Merge Sort): Массив делится пополам, сортируются половины, затем результаты сливаются.

\* Быстрая сортировка (Quick Sort): Массив делится на две части относительно опорного элемента, части сортируются рекурсивно.

\* Бинарный поиск: Пространство поиска делится пополам на каждой итерации.

\* Вычисление числа Фибоначчи (рекурсивный подход, хотя и неэффективный): F(n) = F(n-1) + F(n-2).

• Алгоритмы: Merge Sort, Quick Sort, бинарный поиск, быстрая сортировка, алгоритм Карацубы для умножения больших чисел.

**3. Метод динамического программирования:**

Динамическое программирование применяется к задачам, которые могут быть разбиты на подзадачи с перекрывающимися подпроблемами. Вместо того чтобы решать одни и те же подпроблемы многократно, как в рекурсивных алгоритмах «разделяй и властвуй», динамическое программирование хранит решения подзадач и использует их повторно. Это существенно повышает эффективность. Обычно используется подход "снизу-вверх" (итеративный) или "сверху-вниз" (рекурсивный с мемоизацией).

• Примеры задач:

\* Нахождение кратчайшего пути (алгоритм Флойда-Уоршелла): Кратчайший путь между двумя вершинами строится из кратчайших путей к промежуточным вершинам.

\* Задача о рюкзаке: Оптимизация выбора предметов для заполнения рюкзака с ограниченной вместимостью.

\* Вычисление числа Фибоначчи (эффективный подход): Результаты вычислений F(n-1) и F(n-2) сохраняются.

\* Задача о наибольшей возрастающей подпоследовательности.

**4. Жадный метод(алгоритм)**

Жадный алгоритм на каждом шаге делает локально оптимальный выбор, надеясь, что это приведет к глобально оптимальному решению. Однако жадные алгоритмы не всегда находят оптимальное решение, хотя часто дают достаточно хорошее приближение.

• Примеры задач:

\* Алгоритм Хаффмана для кодирования: На каждом шаге объединяются два узла с наименьшими вероятностями.

\* Алгоритм Крускала для поиска минимального остовного дерева: На каждом шаге добавляется ребро с минимальным весом, не образующее цикла.

\* Выбор задач для выполнения жадным планировщиком: выбирается задача с наименьшим временем выполнения или наибольшим приоритетом.

**5. Метод ветвей и границ**

Метод ветвей и границ — это метод поиска решения в пространстве поиска, который использует стратегию "отсечения" неприемлемых ветвей. Он систематически исследует пространство решений, используя оценки (границы), чтобы исключить участки пространства, которые гарантированно не содержат оптимального решения. Часто используется для задач оптимизации

• Примеры задач:

\* Задача коммивояжера: Поиск кратчайшего маршрута, проходящего через все города ровно один раз.

\* Задача о назначении: Нахождение оптимального сопоставления заданий работникам.

\* Задача о рюкзаке (также может решаться этим методом): Найти оптимальный набор предметов.

## Задание №2

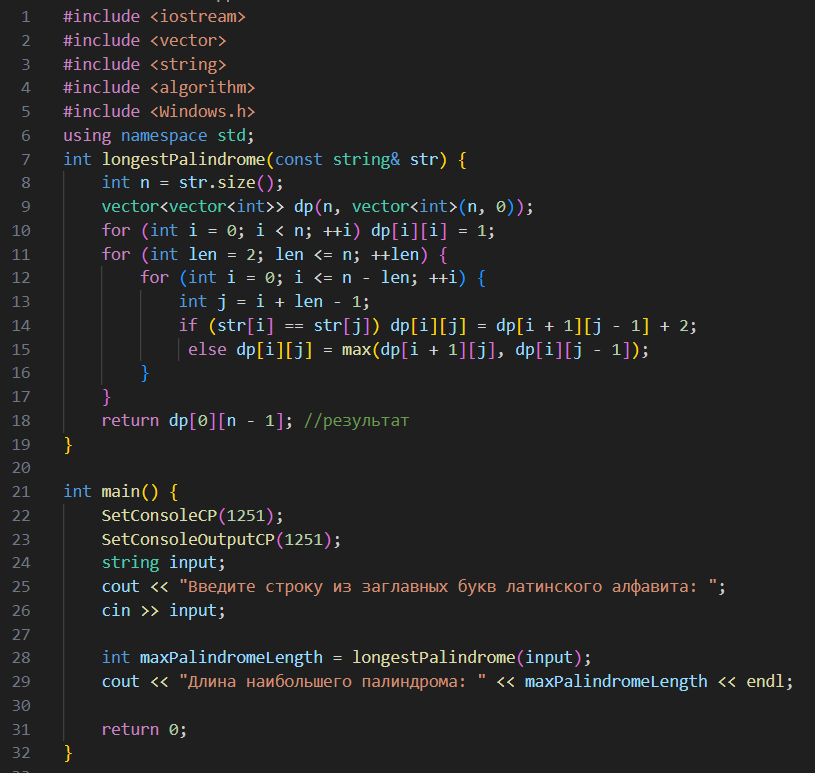


Рисунок 1– код программы

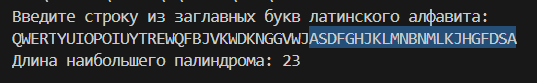


Рисунок 2 - Результат работы программы

# Вывод

В результате выполнения работы и ответов на вопросы я получил практические навыки по выполнению операций над структурой данных граф (дата обращения 09.12.2024).