Пермский национальный исследовательский политехнический университет

Кафедра информационные технологии и автоматизированные системы

Лабораторная работа

Реализация алгоритма решения головоломки «Ханойская башня»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы: ИВТ-24-2б  Ваулин Артём Сергеевич  Проверил:  доцент кафедры ИТАС  О.А. Полякова |

Пермь 2025

# Постановка задачи

**Цель работы:**

1. Разработать и реализовать программный алгоритм для решения классической головоломки "Ханойская башня".
2. Продемонстрировать понимание принципов рекурсивного программирования на примере решения данной задачи.
3. Провести анализ разработанного алгоритма и оценить его эффективность.

**Задание:**

Разработать программу на языке C++, которая решает головоломку "Ханойская башня" для заданного количества дисков. Программа должна:

1. **Принимать на вход количество дисков (N)** от пользователя.
2. **Проверять корректность ввода:** Количество дисков должно быть положительным целым числом. В случае некорректного ввода программа должна выводить сообщение об ошибке.
3. **Реализовывать рекурсивный алгоритм** для решения головоломки.
4. **Выводить на экран последовательность шагов** для перемещения дисков, указывая, какой диск и с какого стержня на какой стержень перемещается на каждом шаге.
5. **Обеспечить корректную работу алгоритма** для любого положительного целого числа дисков (в пределах разумных ограничений, связанных с ресурсами компьютера, например, глубиной рекурсии).
6. **(Дополнительно):** Рассчитать и вывести общее количество ходов, выполненных алгоритмом.

**Входные данные:**

* Целое положительное число N, представляющее количество дисков в головоломке.

**Выходные данные:**

* Последовательность шагов для решения головоломки в формате: "Переместить диск X с стержня Y на стержень Z", где X - номер диска, Y - исходный стержень, Z - целевой стержень.
* (Дополнительно): Общее количество ходов.
* В случае некорректного ввода - сообщение об ошибке.

**Ограничения:**

* Количество дисков N должно быть положительным целым числом.
* Программа должна быть реализована на языке C++.
* Использование рекурсивного подхода для решения головоломки является обязательным.

# Анализ задачи

**1. Декомпозиция задачи и рекурсивный подход:**

Задача "Ханойская башня" обладает ярко выраженной рекурсивной структурой, что делает рекурсивный подход естественным и эффективным методом ее решения. Суть рекурсивного решения заключается в декомпозиции исходной задачи (перемещение N дисков) на более простые подзадачи того же типа:

* **Базовый случай:** Если количество дисков равно 1, задача тривиальна - нужно просто переместить единственный диск с исходного стержня на целевой.
* **Рекурсивный шаг:** Для N > 1 дисков, задача сводится к следующим трем шагам:
  1. **Переместить (N-1) верхних дисков** с исходного стержня на вспомогательный стержень, используя целевой стержень как вспомогательный для этой подзадачи.
  2. **Переместить самый большой (N-й) диск** с исходного стержня на целевой стержень. Это прямой и простой шаг.
  3. **Переместить (N-1) дисков**, которые были перемещены на вспомогательный стержень на шаге 1, на целевой стержень, используя исходный стержень как вспомогательный для этой подзадачи.

Этот рекурсивный процесс продолжается до тех пор, пока не достигается базовый случай (1 диск). Рекурсия позволяет элегантно и компактно выразить алгоритм решения, избегая громоздких итеративных конструкций.

**3. Выбор алгоритма и его обоснование:**

Для решения головоломки "Ханойская башня" рекурсивный алгоритм является наиболее **естественным, интуитивно понятным и каноническим** подходом. Его преимущества:

* **Простота реализации:** Код рекурсивного решения получается коротким, ясным и легко читаемым.
* **Соответствие рекурсивной природе задачи:** Алгоритм непосредственно отражает рекурсивную структуру самой головоломки, что делает его логичным и понятным.
* **Оптимальность (по количеству ходов):** Рекурсивный алгоритм гарантированно находит решение за минимальное количество ходов.

Хотя существуют и итеративные решения для Ханойской башни, они, как правило, значительно сложнее для понимания и реализации, особенно для начинающих программистов. В контексте лабораторной работы, где целью является демонстрация понимания рекурсии, рекурсивный алгоритм является **наиболее подходящим и рекомендуемым выбором**.

# Блок схема

начало

#include <iostream>

#include <locale.h>

void solveHanoi(int numDisks, char sourceRod, char targetRod, char auxiliaryRod)

if (numDisks == 1)

cout << "Переместить диск 1 с стержня " << sourceRod << " на стержень " << targetRod << endl;

return;

solveHanoi(numDisks - 1, sourceRod, auxiliaryRod, targetRod);

cout << "Переместить диск " << numDisks << " с стержня " << sourceRod << " на стержень " << targetRod << endl;

solveHanoi(numDisks - 1, auxiliaryRod, targetRod, sourceRod);

{

{

}

}

1

0

int main()

setlocale(LC\_ALL, "ru\_RU");

int numDisks;

cout << "--- Решение Ханойской башни ---" << endl;

cout << "Введите количество дисков: ";

cin >> numDisks;

if (numDisks <= 0)

cerr << "Ошибка: Количество дисков должно быть положительным целым числом." << endl;

return 1;

cout << endl << "Действия для " << numDisks << " дисков:" << endl;

А

{

{

}

1

0

А

solveHanoi(numDisks, 'A', 'C', 'B');

cout << endl << "--- Решение завершено ---" << endl;

return 0;

}

# Код программы

#include <iostream>

#include <locale.h>

using namespace std;

// Рекурсивное решение головоломки "Ханойская башня".

void solveHanoi(int numDisks, char sourceRod, char targetRod, char auxiliaryRod)

{

if (numDisks == 1)

{

cout << "Переместить диск 1 с стержня " << sourceRod << " на стержень " << targetRod << endl;

return;

}

// 1. Переместить (n-1) дисков со source на auxiliary

solveHanoi(numDisks - 1, sourceRod, auxiliaryRod, targetRod);

// 2. Переместить самый большой диск (n) с source на target

cout << "Переместить диск " << numDisks << " с стержня " << sourceRod << " на стержень " << targetRod << endl;

// 3. Переместить (n-1) дисков с auxiliary на target

solveHanoi(numDisks - 1, auxiliaryRod, targetRod, sourceRod);

}

int main()

{

setlocale(LC\_ALL, "ru\_RU");

int numDisks;

cout << "--- Решение Ханойской башни ---" << endl;

cout << "Введите количество дисков: ";

cin >> numDisks;

if (numDisks <= 0)

{

cerr << "Ошибка: Количество дисков должно быть положительным целым числом." << endl;

return 1;

}

cout << endl << "Действия для " << numDisks << " дисков:" << endl;

solveHanoi(numDisks, 'A', 'C', 'B');

cout << endl << "--- Решение завершено ---" << endl;

return 0;

}

# Результат работы

