ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 4

«Задача о согласовании»

Выполнил работу Мельников Олег Академическая группа №J3112

Принято Дунаев Максим Владимирович

Санкт-Петербург 2024

1. ВВЕДЕНИЕ

**Цель работы:** разработать и реализовать алгоритм на языке программирования C++ для решения задачи о согласовании.

**Задачи:**

* Изучить условия задачи;
* Разработать алгоритм;
* Проанализировать результаты, оценить эффективность.

2. ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА

Задача заключается в следующем: Дан набор пар, необходимо найти максимальное согласование, чтобы каждая пара состояла из несовместимых элементов.

Задача должна быть решена комбинаторным способом  
Сложность алгоритма должна быть не меньше, чем O(2^N)  
Максимум 25 пар и 30 элементов в парах

1. РЕАЛИЗАЦИЯ

* findCombinations: Эта рекурсивная функция находит уникальные значения, добавляя их в unordered\_set. Она обеспечивает, что каждое число встречается только один раз.

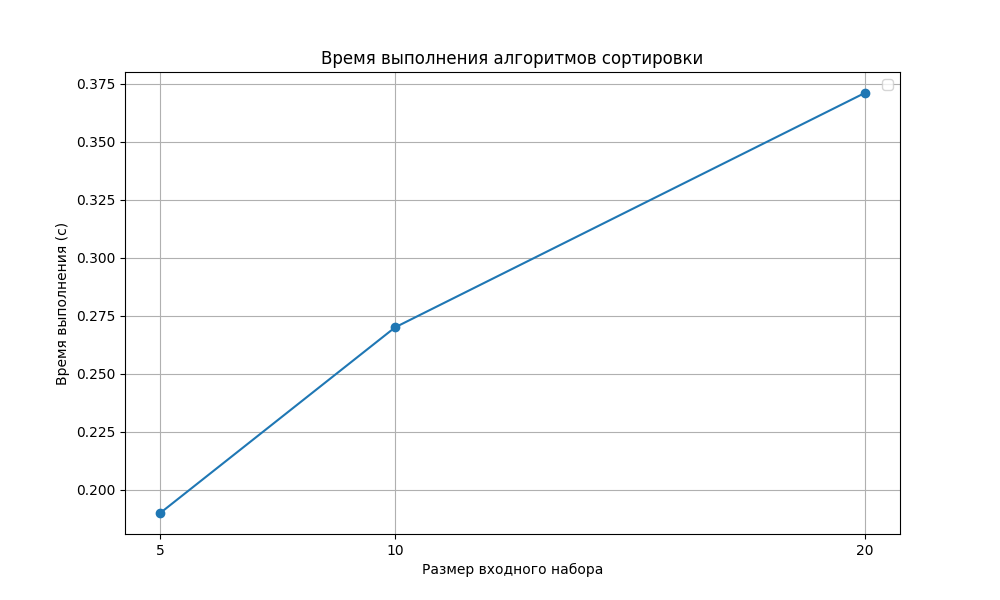
1. ЭКСПЕРЕМИНТАЛЬНАЯ ЧАСТЬ

**Таблица 1 - Подсчет времени выполнения реализованных алгоритмов**

|  |  |  |  |
| --- | --- | --- | --- |
| Размер входных данных (пар) | 5 | 10 | 20 |
| Время | 0.19 | 0.27 | 0.371 |

**График зависимости времени от числа элементов**

График, представляющий время выполнения алгоритма в зависимости от размеров входных данных, представлен ниже.

****

Анализ

**Экспоненциальный рост времени выполнения**:

* Заметим, что время выполнения увеличивается не линейно с увеличением количества пар. Для 5 пар время составляет 0.19 секунд, для 10 пар — 0.27 секунд, а для 20 пар — 0.371 секунд.
* Текущая реализация использует рекурсию для обработки каждой пары. Таким образом, количество вызовов функции findCombinations растет экспоненциально. На каждом уровне рекурсии функция делает два вызова, что приводит к 2^n вызовам для n пар.

1. ЗАКЛЮЧЕНИЕ

В ходе выполнения работы был разработан и реализован алгоритм согласования пар.

Цель работы была достигнута: алгоритм успешно решает задачу на тестовых данных. Также алгоритм имеет сложность O(2^n)

ПРИЛОЖЕНИЕ

Листинг кода решения

#include <iostream>

#include <vector>

#include <unordered\_set>

#include <algorithm>

#include <fstream>

#include <sstream>

#include <ctime>

using namespace std;

void findCombinations(const vector<pair<int, int>>& pairs, size\_t index, unordered\_set<int>& result) {

    if (index >= pairs.size()) {

        return;

    }

    result.insert(pairs[index].first);

    result.insert(pairs[index].second);

    findCombinations(pairs, index + 1, result);

    findCombinations(pairs, index + 1, result);

}

int main() {

    vector<pair<int, int>> pairs;

    ifstream input("C:\\Users\\oleg-\\lr41.txt"); //5 0.19, 10 0.27, 20 0.371

    if (!input) {

        cerr << "Н" << endl;

        return 1;

    }

    string line;

    while (getline(input, line)) {

        stringstream ss(line);

        int first, second;

        ss >> first;

        ss.ignore(1, '\t');

        ss >> second;

        pairs.emplace\_back(first, second);

    }

    input.close();

    cout << "pairs:" << endl;

    for (const auto& p : pairs) {

        cout << "(" << p.first << ", " << p.second << ")" << endl;

    }

    unordered\_set<int> resultSet;

    findCombinations(pairs, 0, resultSet);

    vector<int> vec(resultSet.begin(), resultSet.end());

    cout << "[";

    for (auto it = vec.rbegin(); it != vec.rend(); ++it) {

        cout << \*it << " ";

    }

    cout << "]" << endl;

    cout << "runtime = " << clock()/1000.0 << endl;

    return 0;

}