



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по домашнему заданию
по курсу «Анализ Алгоритмов»
на тему: «Графовые представления»

Студент группы ИУ7-54Б

(Подпись, дата)

Спирин М. П.

(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Волкова Л. Л.

(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Строганов Ю. В..

(Фамилия И.О.)

Москва — 2023 г.

Содержание

Введение	3
1 Аналитическая часть	4
1.1 Реализация алгоритма поиска подстроки в файле	4
1.2 Графовые представления	6
1.3 Возможность распараллеливания	6
Заключение	11

Введение

Целью лабораторной является описание алгоритма четырьмя графовыми моделями (граф управления, информационный граф, операционная история, информационная история).

Для поставленной цели необходимо выполнить следующие задачи.

- 1) Описать графовые модели.
- 2) Описать алгоритм с помощью графовых моделей.
- 3) Предложить вариант распараллеливания.

Вариант: реализуемый алгоритм — поиск всех вхождений подстроки в файл.

1 Аналитическая часть

1.1 Реализация алгоритма поиска подстроки в файле

В листинге 1.1 приведена реализация алгоритма поиска подстроки в строках, читаемых из файла. При нахождении подстроки, в результирующий файл записываются номер строки и столбца первого символа найденной подстроки.

Листинг 1.1 – Функция поиска подстроки в файле с последующей записью в результирующем файле

```
1 in_file — название исходного файла           // -3
2 out_file — название результирующего файла     // -2
3 substr — искомая подстрока                    // -1
4
5
6 void substr_search_pos(std::string &in_file , std::string
   &out_file , std::string &substr)
7 {
8     std::ifstream in(in_file , std::ios::in);   // 1
9     if (!in.is_open())                          // 2
10    {
11        return;                                  // 3
12    }
13    std::ofstream out(out_file);                 // 4
14    if (!out.is_open())                          // 5
15    {
16        return;                                  // 6
17    }
18    std::string line;                             // 7
19    size_t pos;                                   // 8
20    int i = 0;                                    // 9
21    while (getline(in , line)){                   // 10
22        pos = line.find(substr);                  // 11
23        while (pos != std::string::npos)          // 12
24        {
25            out << i + 1 << " " << pos << "\n"; // 13
26            pos = line.find(substr , pos + 1);    // 14
27            i++;                                  // 15
28        }
29    }
30    in.close();                                   // 16
31    out.close();                                  // 17
32 }
```

1.2 Графовые представления

На рисунках 1.1 – 1.4 показаны граф управления, информационный граф, операционная история и информационная история соответственно.

1.3 Возможность распараллеливания

Алгоритм поиска подстроки в файле можно распараллелить, выделив 3 вида потоков:

- поток-читатель — последовательно читает строки из файла;
- поток-обработчик — получает одну из прочитанных строк и находит все вхождения подстроки в неё;
- поток-писатель — получает результаты обработки и записывает их в результирующий файл.

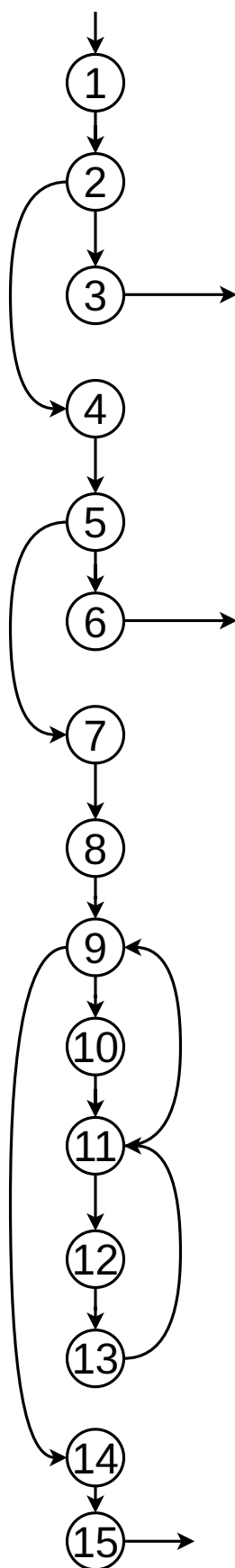


Рисунок 1.1 – Операционный граф

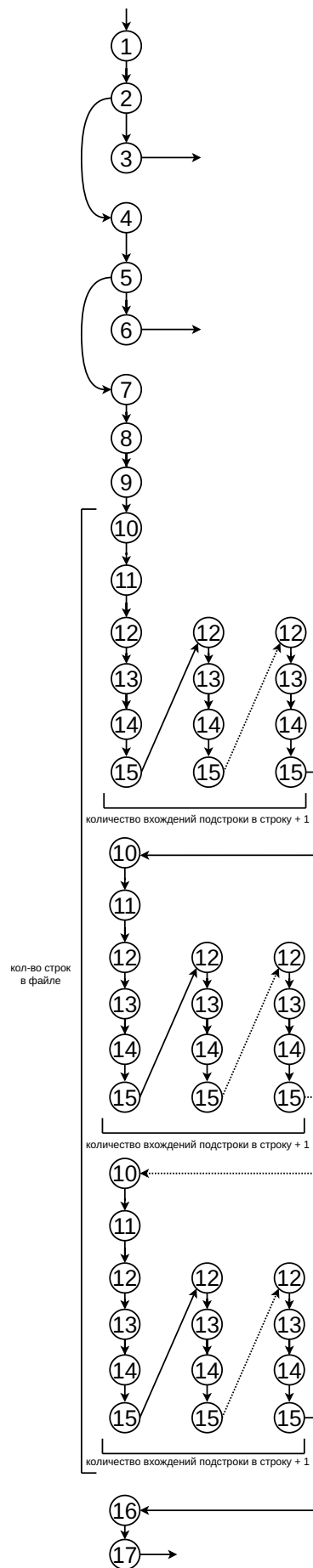


Рисунок 1.3 – Операционная история

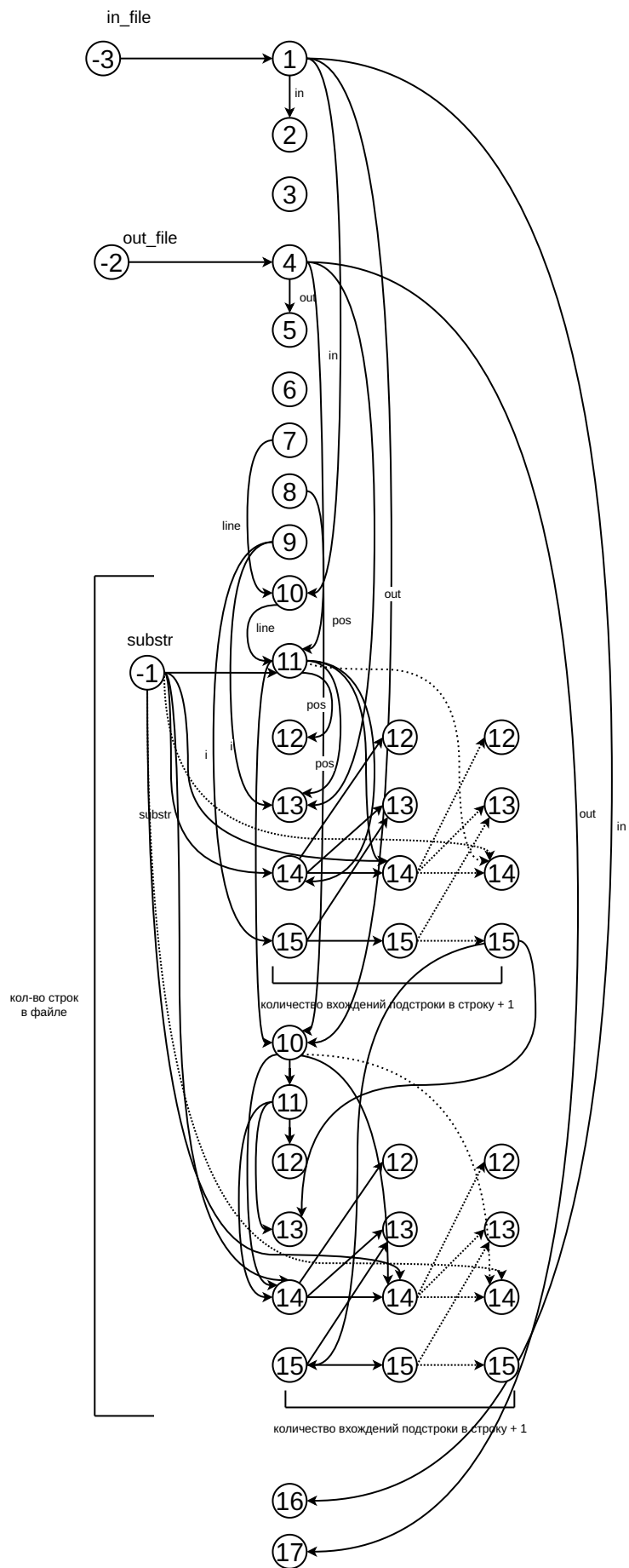


Рисунок 1.4 – Информационная история

Заключение

Поставленная цель была достигнута: алгоритм был описан четырьмя графовыми моделями (граф управления, информационный граф, операционная история, информационная история).

В ходе выполнения лабораторной работы были решены все задачи:

- 1) описаны графовые модели;
- 2) описан алгоритм с помощью графовых моделей;
- 3) предложен вариант распараллеливания.