



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 Лабораторной работе №4

по курсу «Анализы Алгоритмов»

на тему: «Параллельные вычисления на основе нативных  
ПОТОКОВ»

Студент группы ИУ7-56Б

\_\_\_\_\_  
(Подпись, дата)

Мансуров В. М.

\_\_\_\_\_  
(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

<b>1 Выполнения задания</b>	<b>3</b>
1.1 Реализация алгоритма расчета TF для выборки документов	3
1.2 Графовые представления . . . . .	4

# **1 Выполнения задания**

## **1.1 Реализация алгоритма расчета TF для выборки документов**

В листинге 1.1 приведены реализации алгоритма выделения расчета термовой частоты для каждого терма из выборки документов. В качестве термов в данной реализации рассматриваются слова, состоящие из латинских букв. В качестве документов рассматриваются строки, состоящие из таких слов, пробелов и знаков пунктуации.

Листинг 1.1 – Функция реализации алгоритма расчета термовой частоты для каждого терма из выборки документов

```
1 docs_cnt — количество документов // -3
2 ctn — результирующая матрицы // -2
3 docs — массив документов // -1
4
5 void find_ctn_tf(mtr_t *docs, ctn_t *ctn, size_t docs_cnt)
6 {
7     for (int id = 0; id < docs_cnt; id++) // 1
8     {
9         ctn[id]->n = 0; // 2
10        int word_cnt; // 3
11        for (int i = 0, k = 0; i < docs[id]->n; i++) // 4
12        {
13            if (check_word_in_dict(ctn[id]->buff,
14                                   docs[id]->buff[i],
15                                   ctn[id]->n)) // 5
16                continue; // 6
17            word_cnt = 1; // 7
18            for (int j = i + 1; j < docs[id]->n; j++) // 8
19                if (strcmp(docs[id]->buff[i],
20                           docs[id]->buff[j]) == 0) // 9
21                    word_cnt++; // 10
22
23            strcpy(ctn[id]->buff[k].word,
24                   docs[id]->buff[i]); // 11
25            ctn[id]->buff[k].cnt = tf(word_cnt, docs[id]->n); // 12
26            ctn[id]->n++; // 13
27            k++; // 14
28        }
29    }
30 }
```

## 1.2 Графовые представления

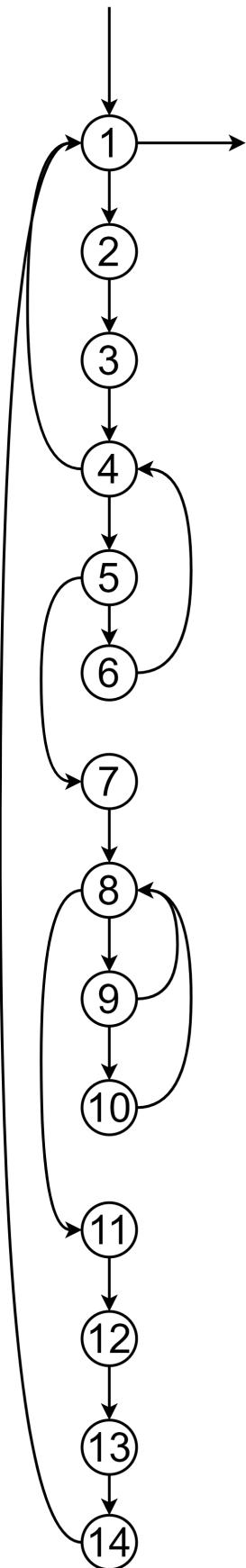


Рисунок 1.1 – Операционный граф

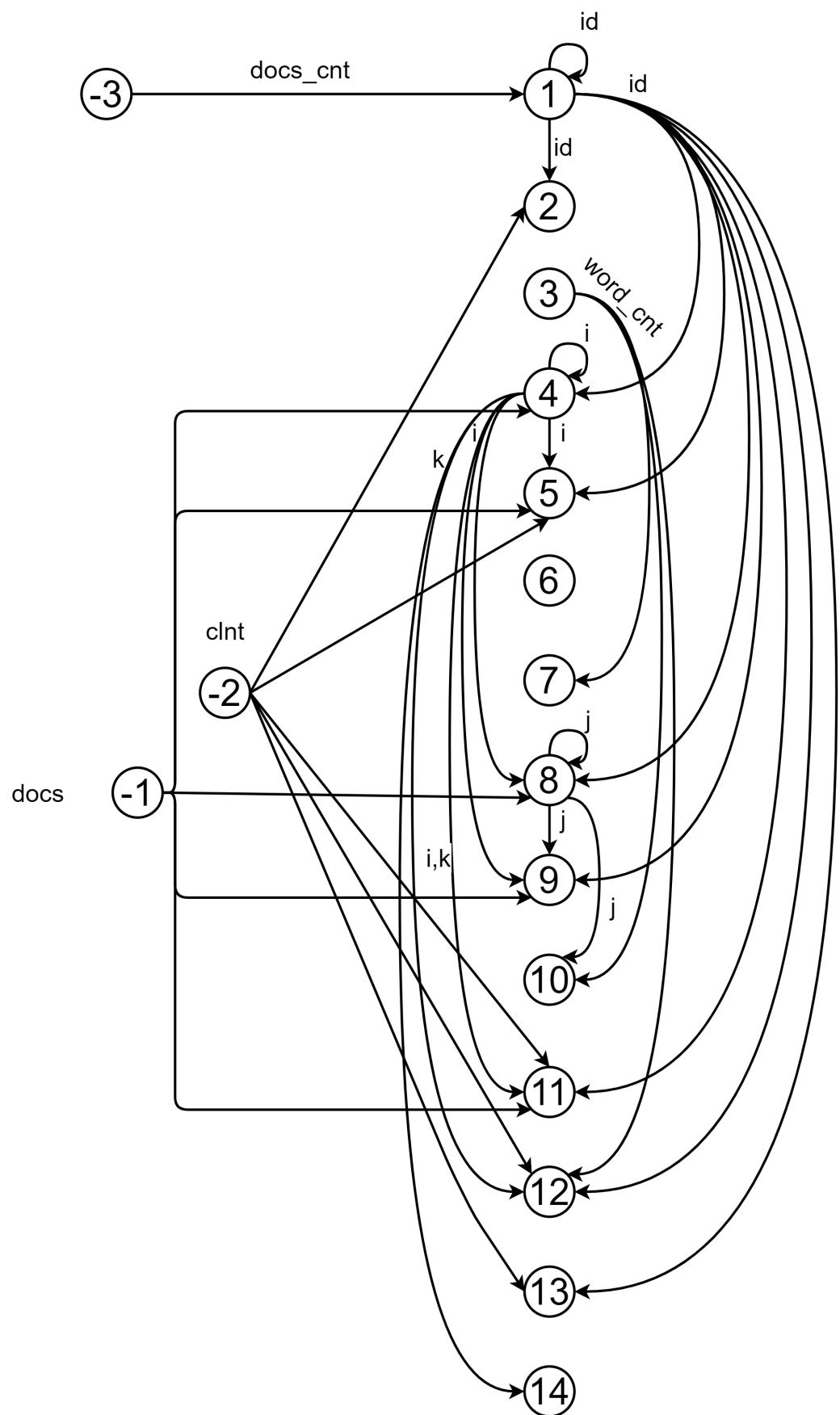


Рисунок 1.2 – Информационный граф

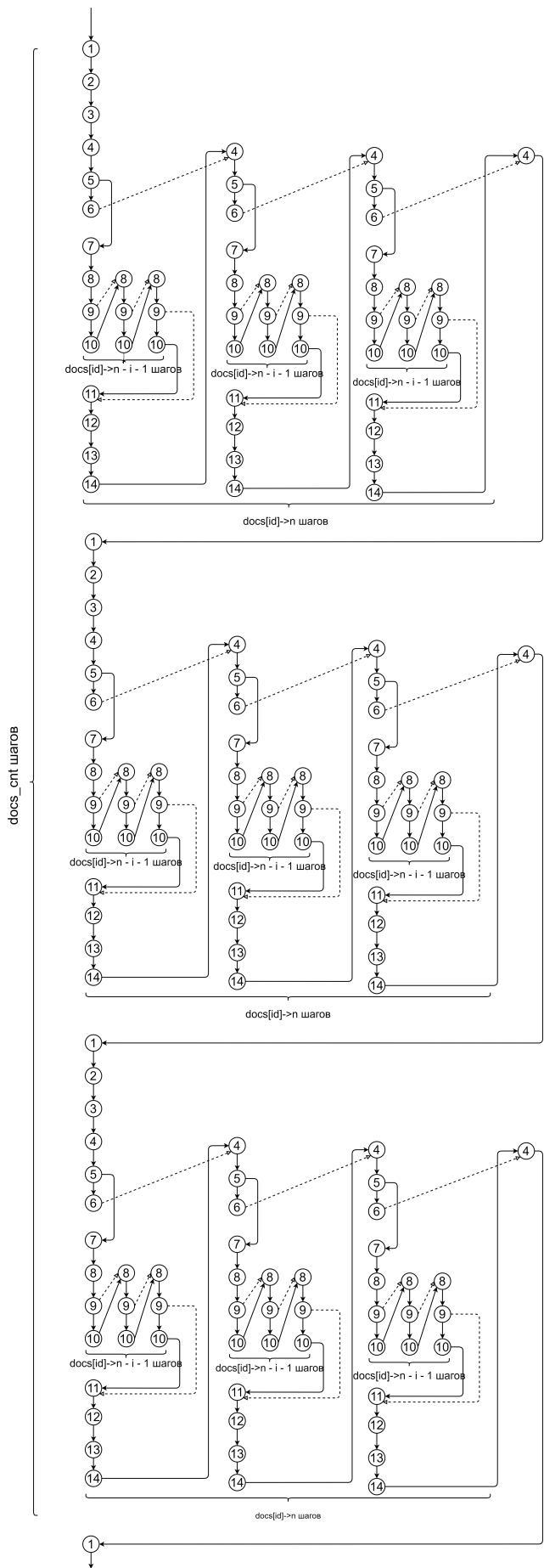


Рисунок 1.3 – Операционная история

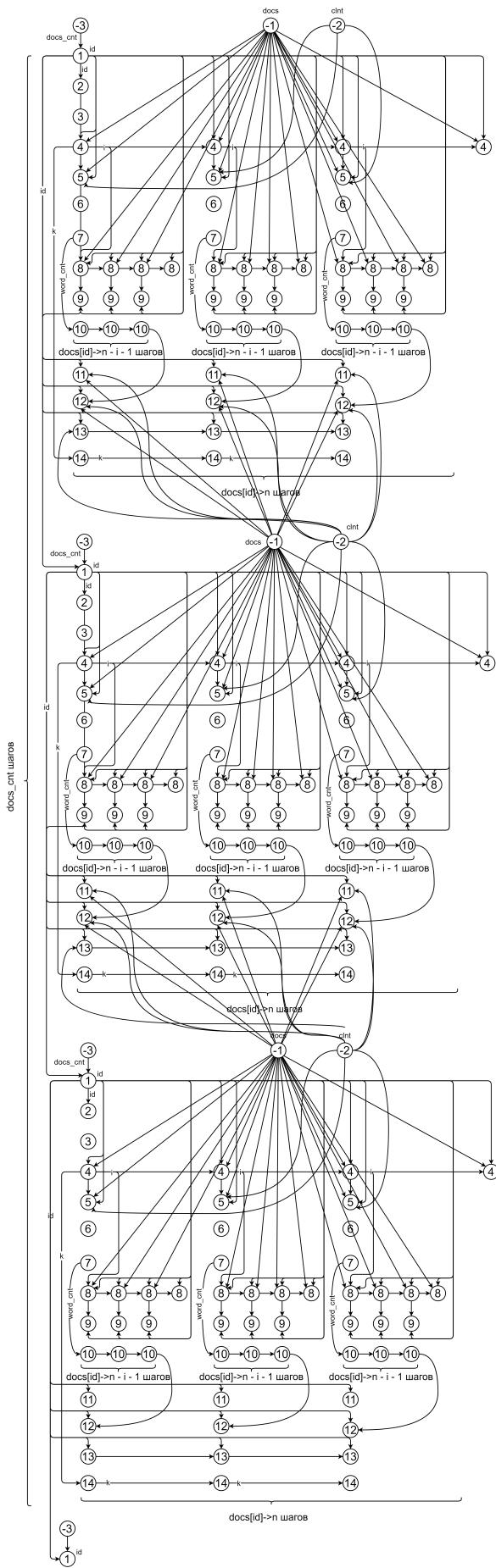


Рисунок 1.4 – Информационная история