



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 модели программ»

Студент группы ИУ7-51Б

\_\_\_\_\_  
(Подпись, дата)

Постнов С. А.  
\_\_\_\_\_  
(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</b>	<b>Выполнение задания</b>	<b>3</b>
1.1	Реализуемый алгоритм . . . . .	3
1.2	Выбор языка программирования . . . . .	3
1.3	Код программы . . . . .	3
1.4	Модели программ . . . . .	4
1.4.1	Граф управления программы . . . . .	4
1.4.2	Информационный граф программы . . . . .	5
1.4.3	Операционная история программы . . . . .	6
1.4.4	Информационная история программы . . . . .	7
1.4.5	Возможность распараллеливания . . . . .	7

# 1 Выполнение задания

## 1.1 Реализуемый алгоритм

Расчет значения полинома по схеме Горнера.

## 1.2 Выбор языка программирования

Для выполнения домашнего задания был выбран язык C++.

## 1.3 Код программы

Листинг 1.1 – Реализация алгоритма расчета значения полинома по схеме Горнера

```
1 int main() {  
2     srand(time(0));  
3  
4     int power = 5; // (1)  
5     arrayT<double> koefs(power + 1); // (2)  
6  
7     for (int i = 0; i <= power; ++i) { // (3)  
8         koefs[i] = rand() % 100 - 50; // (4)  
9         cout << "Коэффициента при x^" // (5)  
10            << power - i << ": "  
11            << koefs[i] << endl;  
12     }  
13  
14     arrayT<double> tmp(power + 1); // (6)  
15     double x = 1; // (7)  
16  
17     for (int i = 0; i <= power; ++i) { // (8)  
18  
19         tmp[i] = koefs[i]; // (9)  
20  
21         if (i) // (10)
```

```

22         tmp[i] += x * tmp[i - 1];           // (11)
23     }
24
25     cout << "\nРезультат: " << tmp[power]; // (12)
26 }

```

## 1.4 Модели программ

### 1.4.1 Граф управления программы

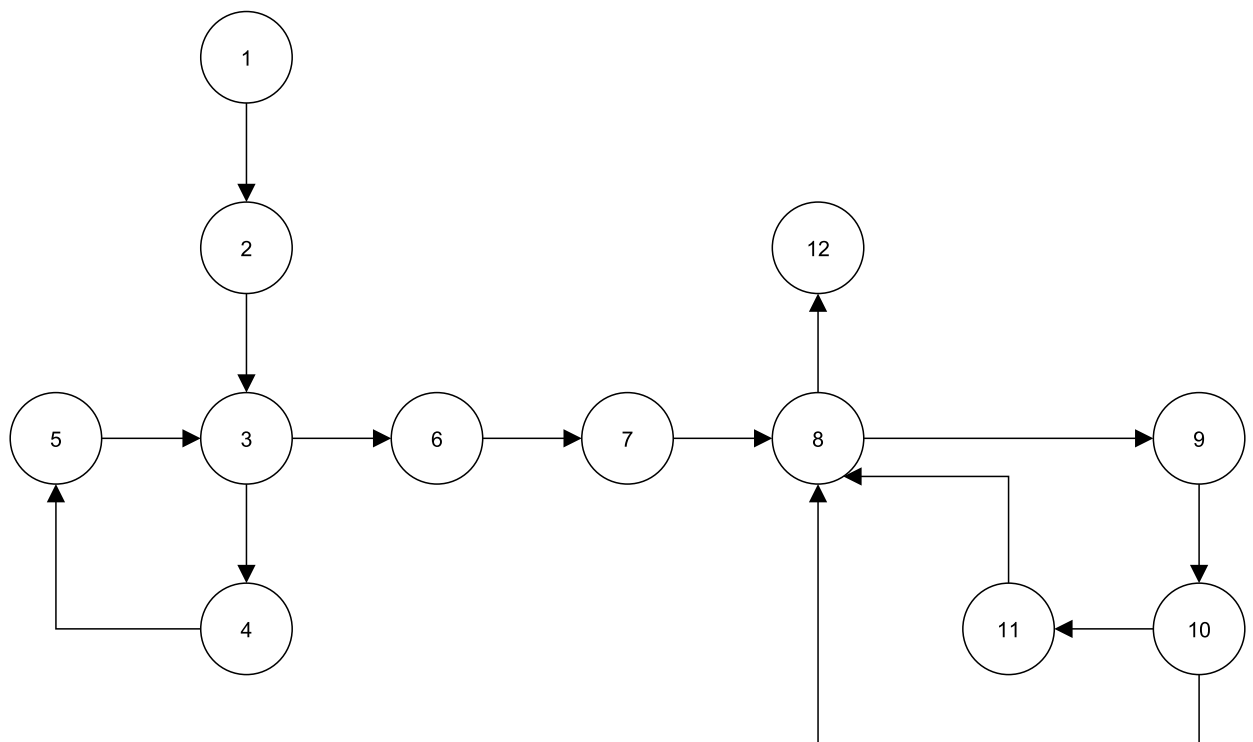


Рисунок 1.1 – Граф управления

### 1.4.2 Информационный граф программы

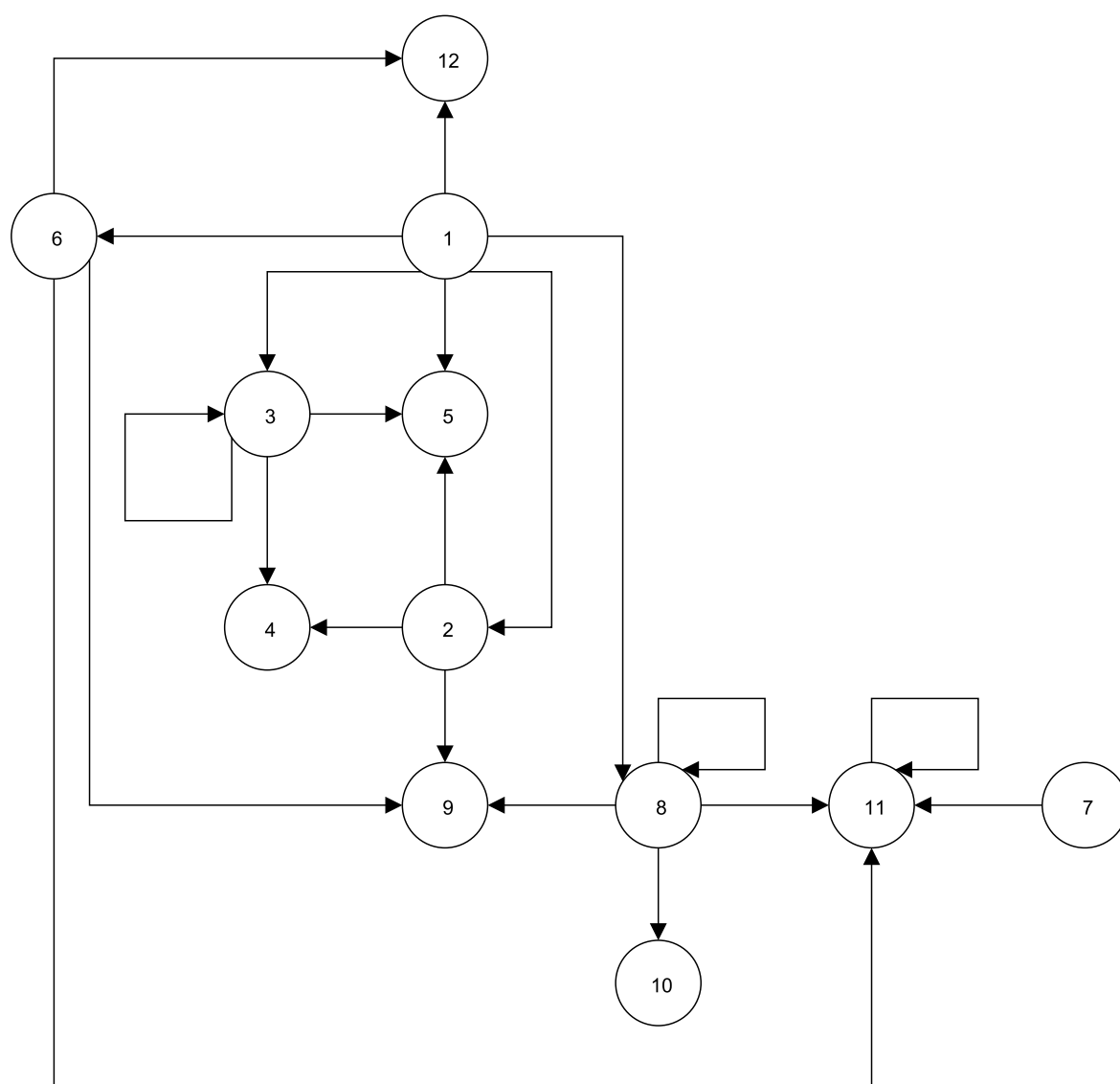


Рисунок 1.2 – Информационный граф

### 1.4.3 Операционная история программы

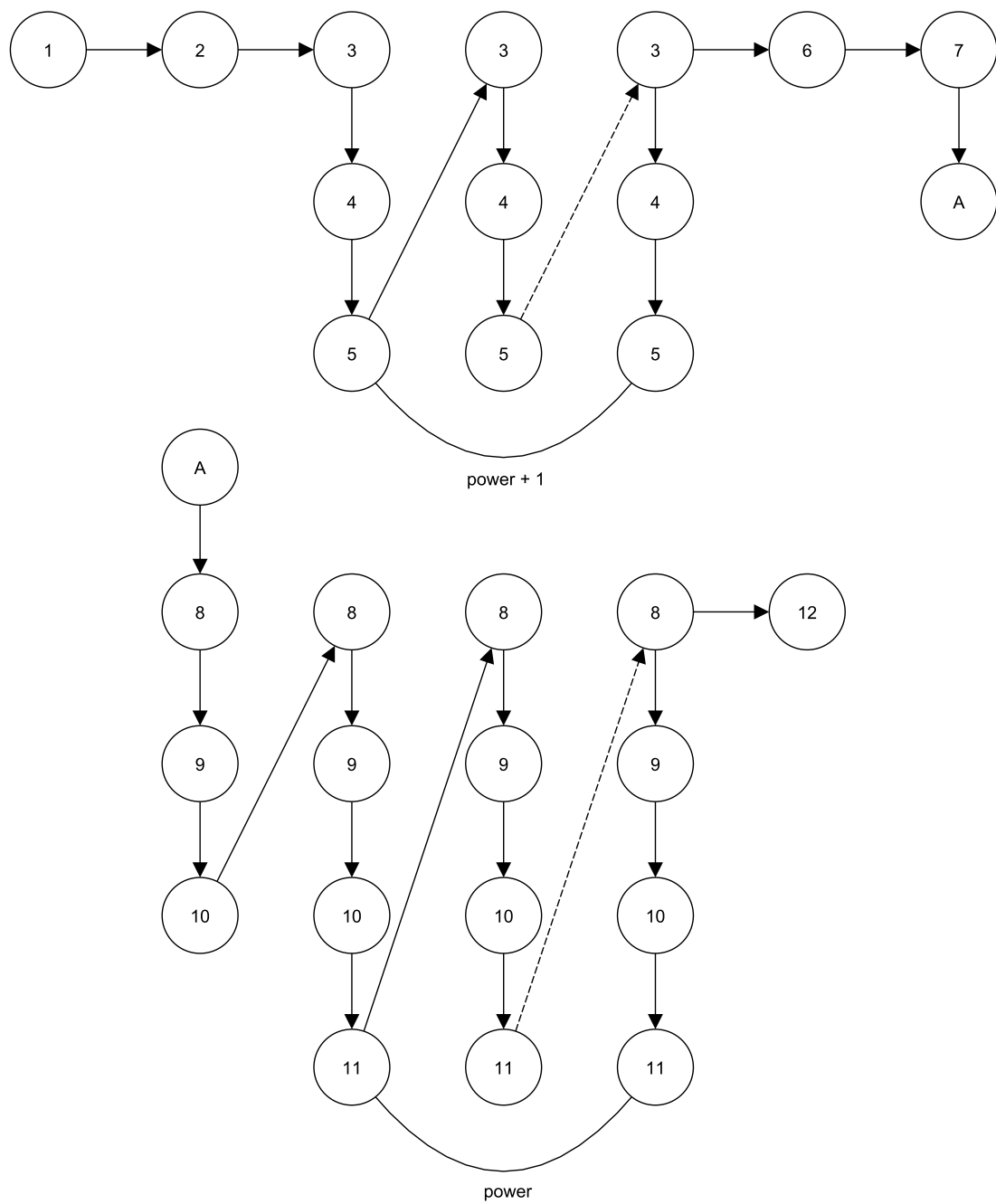


Рисунок 1.3 – Операционная история

#### 1.4.4 Информационная история программы

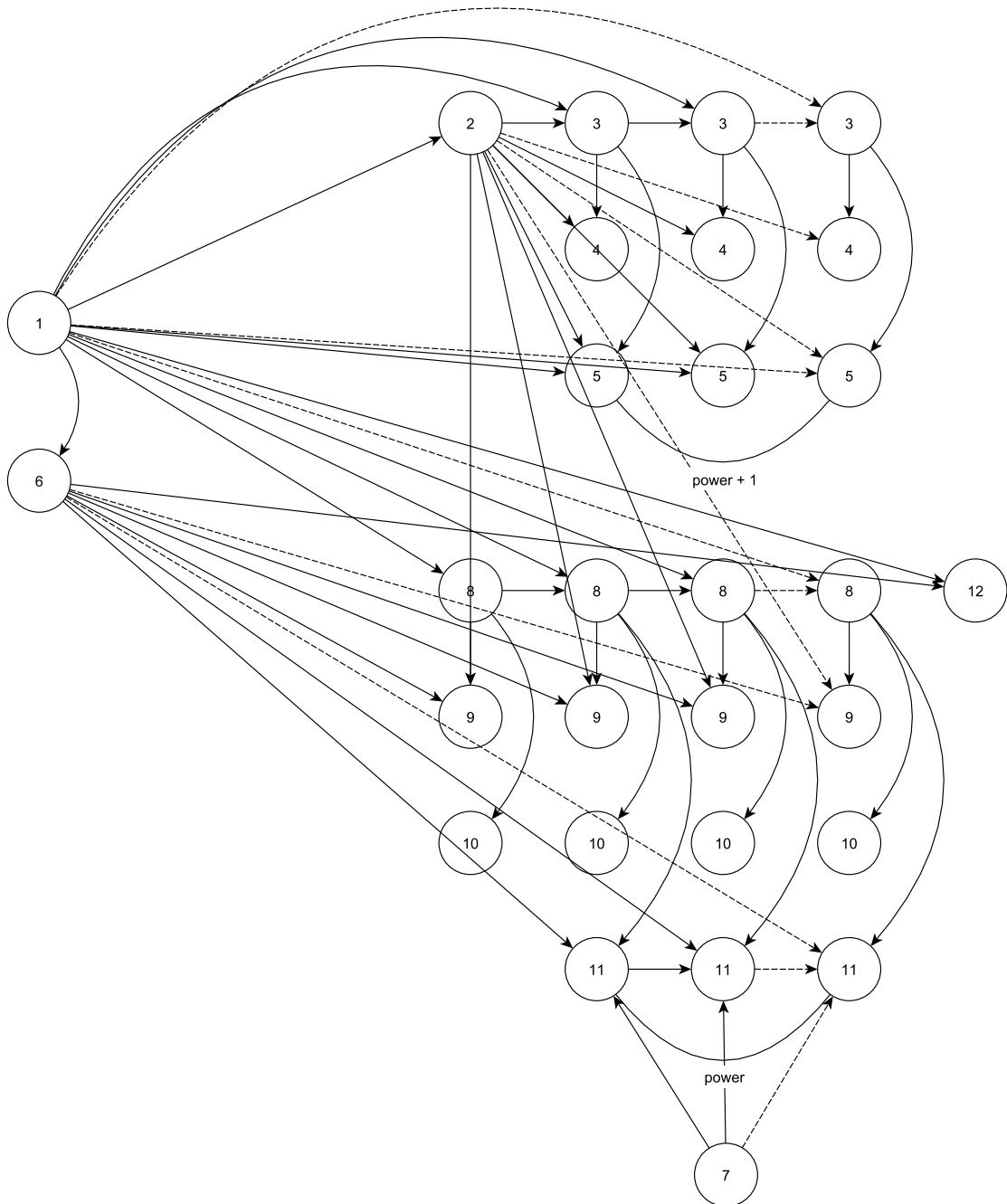


Рисунок 1.4 – Информационная история

#### 1.4.5 Возможность распараллеливания

Один из способов — разделить вычисления на несколько частей, каждую из которых будет обрабатывать свой поток. Например, можно разде-

лить массив коэффициентов на равные подмассивы и запустить вычисление частей полинома в отдельных потоках. То есть каждый поток будет обрабатывать новый подмассив как отдельный полином.