



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)

Савинова М. Г.

(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

1	Выполнение задания	3
1.1	Реализуемый алгоритм	3
1.2	Выбор языка программирования	3
1.3	Код программы	3
1.4	Модели программ	5
1.4.1	Граф управления программы	5
1.4.2	Информационный граф программы	6
1.4.3	Операционная история программы	7
1.4.4	Информационная история программы	9

1 Выполнение задания

1.1 Реализуемый алгоритм

Сортировка слиянием

1.2 Выбор языка программирования

Для выполнения домашнего задания был выбран язык C++.

1.3 Код программы

В листинге 1.1 приведена реализация алгоритма сортировки слиянием.

Листинг 1.1 – Реализация алгоритма сортировки слиянием

```
1 int main() {  
2  
3 srand(time(0));  
4  
5 int size = 4;  
   //(1)  
6 arrayT arr(size);  
   //(2)  
7  
8 for (int i = 0; i < size; ++i)  
9     arr[i] = rand() % 100 - 50;  
10  
11 cout << "До: ";  
12 for (auto elem : arr)  
13     cout << elem << " ";  
14 cout << endl;  
15  
16 for (int step = 1; step < size; step = step * 2){  
17     //(3)
```

```

18   for (int left = 0; left < size - 1; left += 2 * step){
19       //(4)
20       int mid = min(left + step - 1, size - 1),
21           //(5)
22           right = min(left + 2 * step - 1, size - 1);
23           //(6)
24       int i = left ,
25           //(7)
26           j = mid + 1,
27           //(8)
28           pos = 0,
29           //(9)
30           len = right - left + 1;
31           //(10)
32       arrayT tmp(len);
33           //(11)
34       while (i <= mid || j <= right) {
35           //(12)
36           if (i <= mid && (j > right || arr[i] < arr[j]))
37               { //(13)
38                   tmp[pos] = arr[i];
39                   //(14)
40                   i++;
41                   //(15)
42               }
43           else {
44               tmp[pos] = arr[j];
45               //(16)
46               j++;
47               //(17)
48           }
49           pos++;
50           //(18)
51       }
52       for (int k = 0; k < len; ++k)

```

```

44         //(19)
           arr[k + left] = tmp[k];
           //(20)
45     }
46 }
47
48     cout << "После: ";
49     for (auto elem : arr)
50         cout << elem << " ";
51     cout << endl;
52 }

```

1.4 Модели программ

На рисунках 1.1–1.6 представлены модели графовых управлений.

1.4.1 Граф управления программы

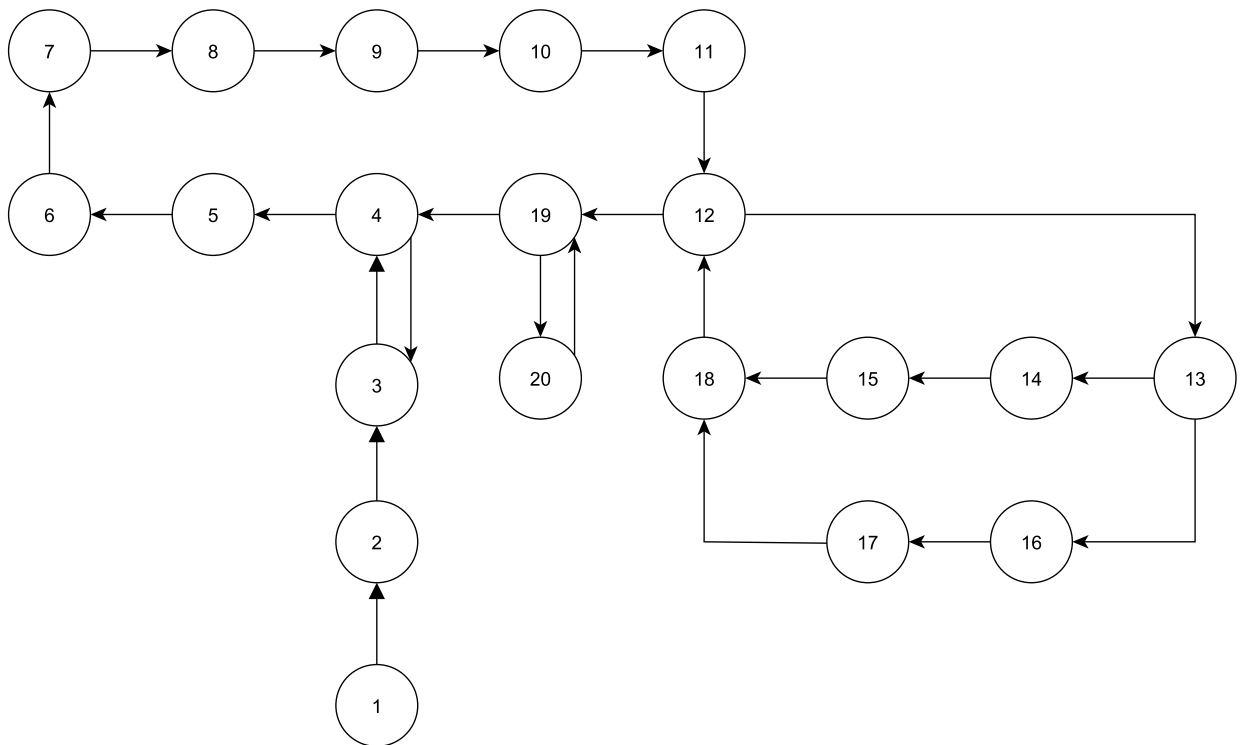


Рисунок 1.1 – Граф управления

1.4.2 Информационный граф программы

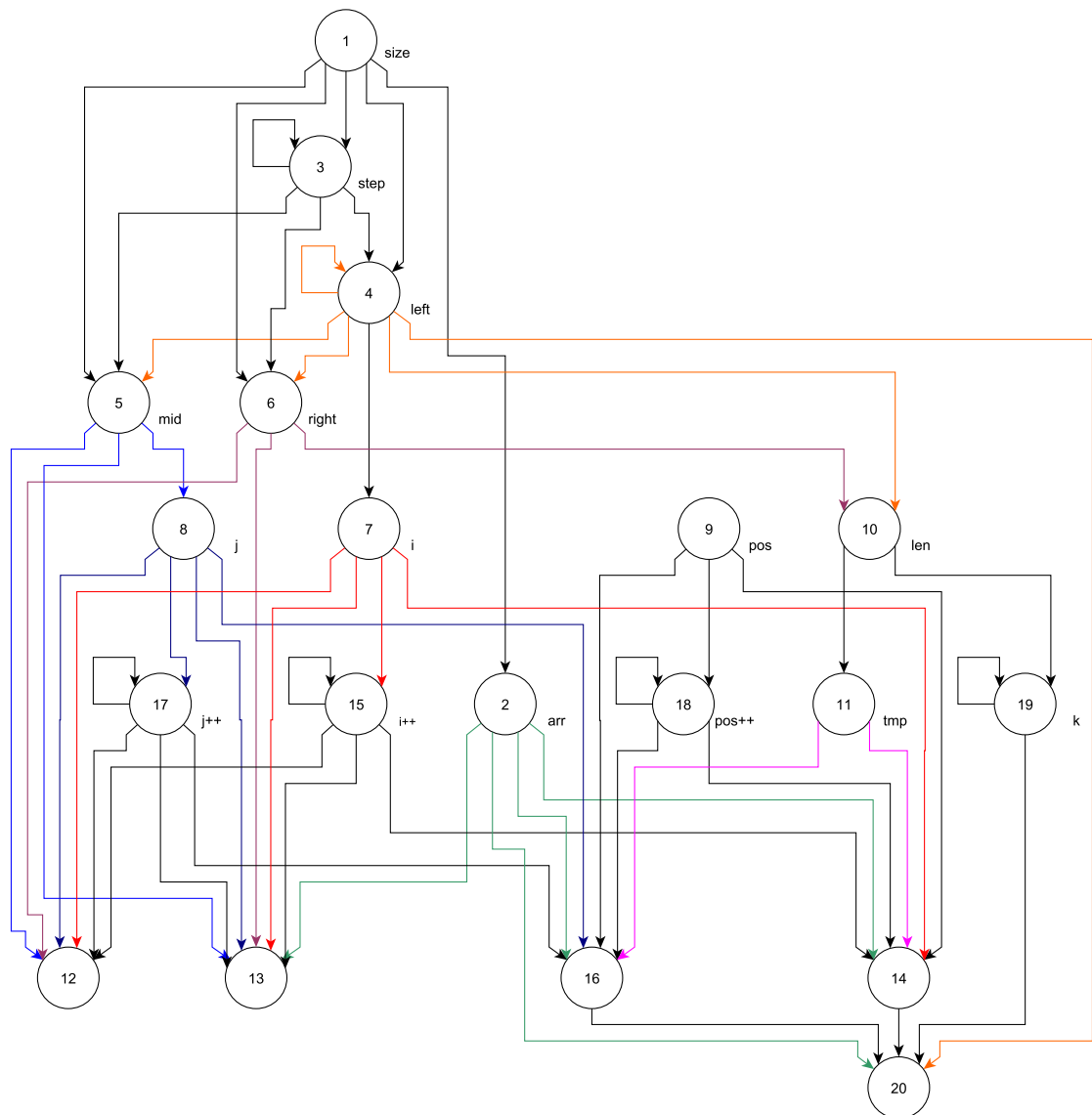


Рисунок 1.2 – Информационный граф

1.4.3 Операционная история программы

Рассмотрим следующий массив: $a = [4, 3, 2, 1]$.

На рисунках 1.3–1.4 представлена операционная история программы для этого массива.

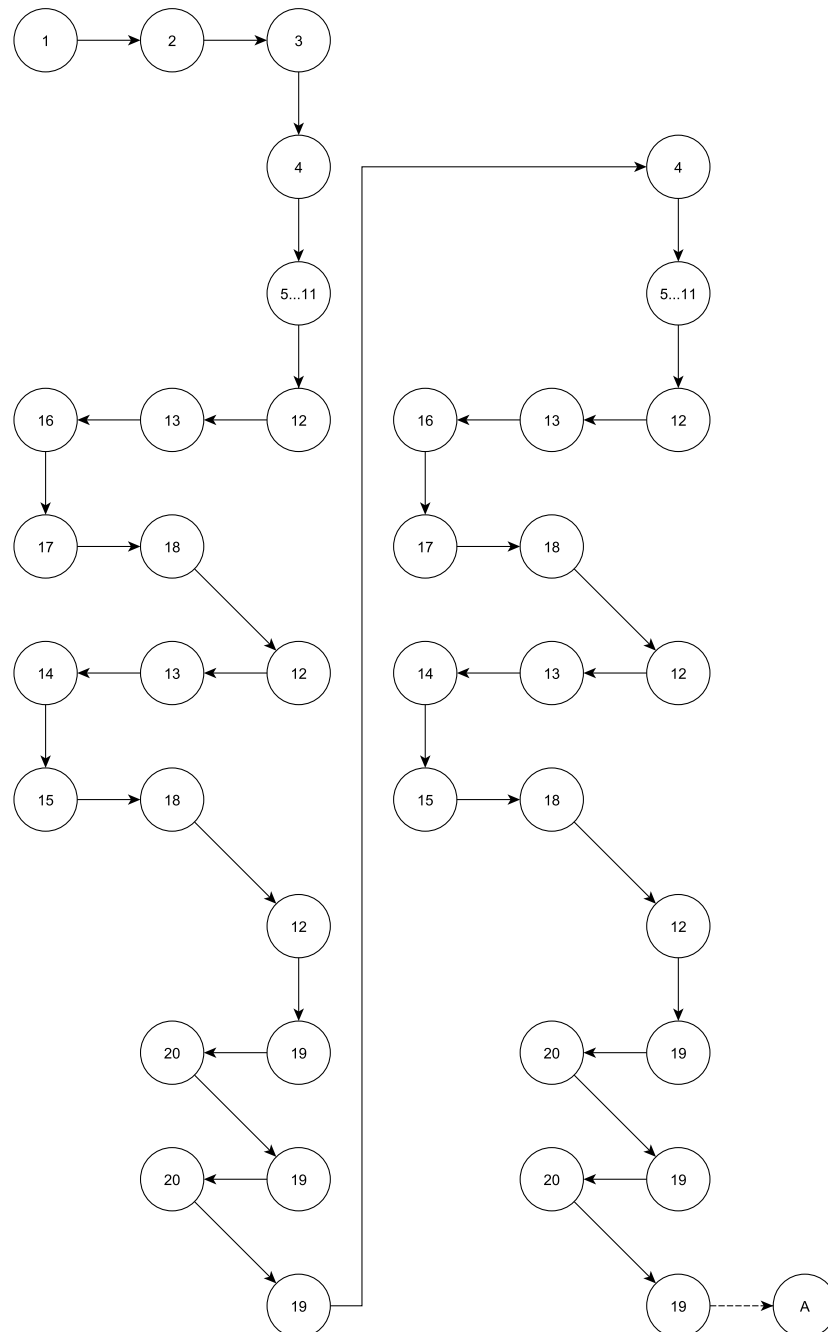


Рисунок 1.3 – Операционная история (начало)

1.4.4 Информационная история программы

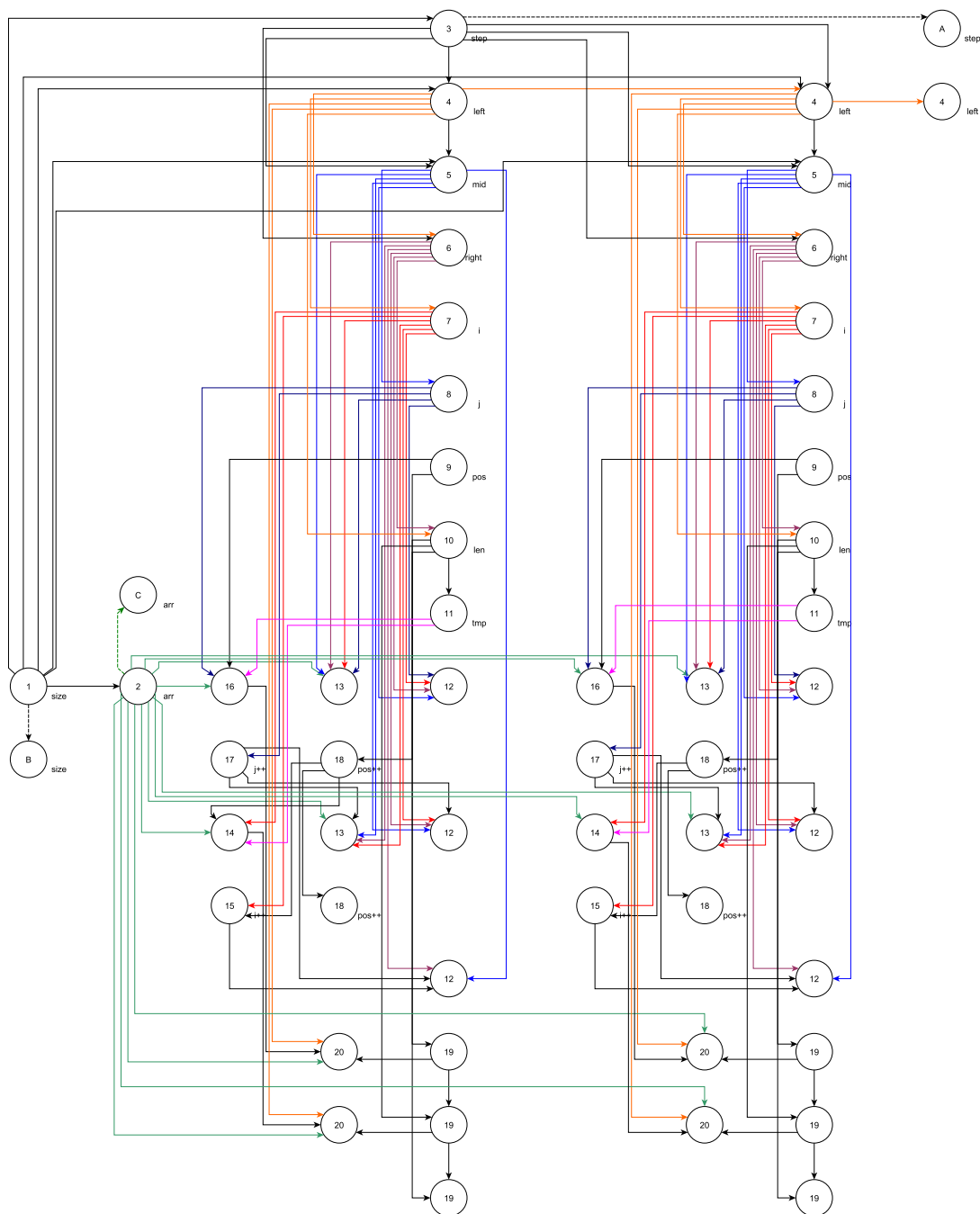


Рисунок 1.5 – Информационная история (начало)

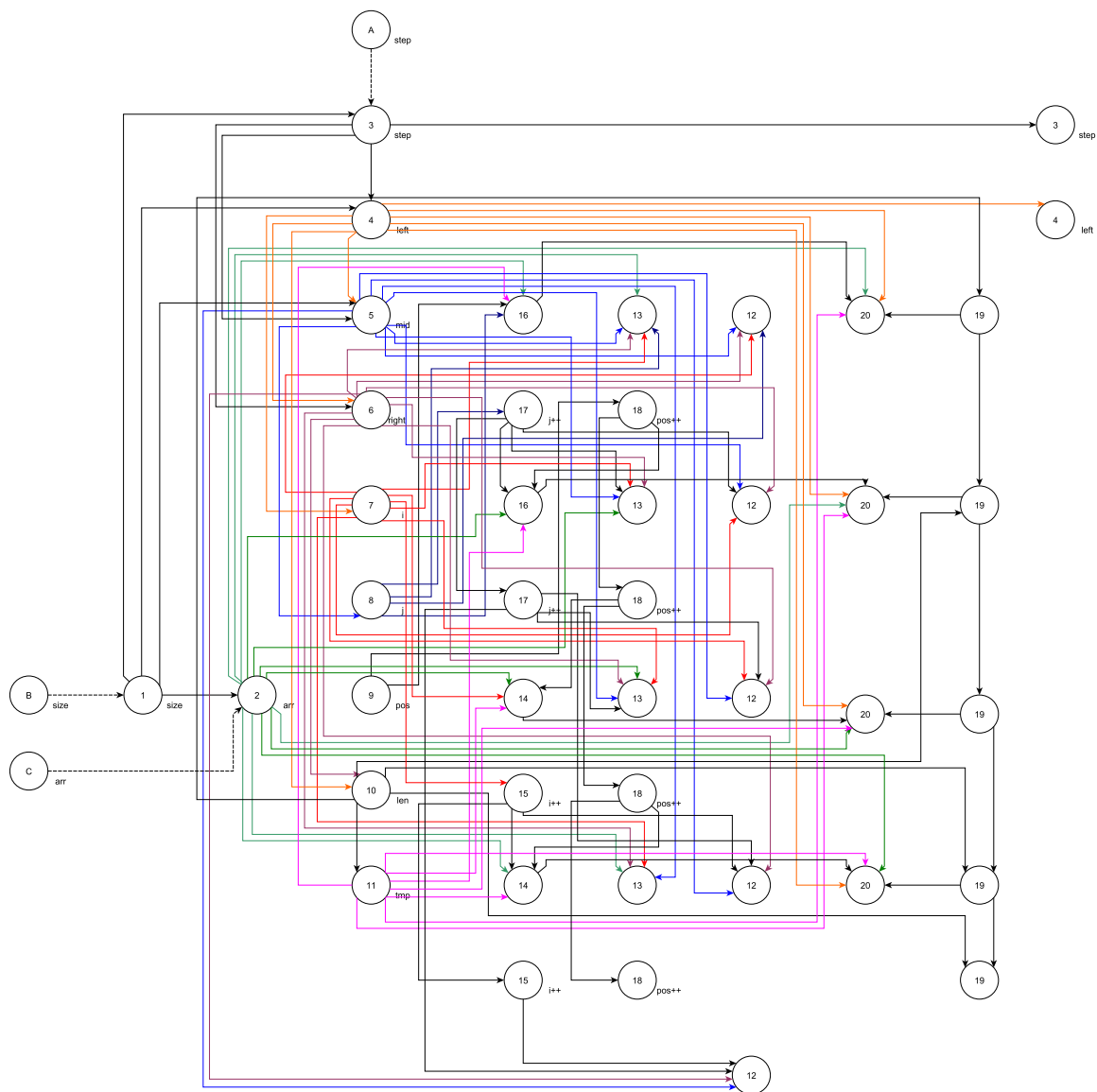


Рисунок 1.6 – Информационная история (конец)

Возможность распараллеливания

Можно разделить массив на части и запустить каждую часть сортировки в отдельном потоке, а затем объединить результаты.