



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	Введение	3
1.1	Задание	3
1.2	Графовые модели программы	3
2	Выполнение	4
2.1	Выбор языка программирования	4
2.2	Код программы	4
2.3	Графовые модели программы	6
2.3.1	Граф управления	6
2.3.2	Информационный граф	7
2.3.3	Операционная история	8
2.3.4	Информационная история	10

1 Введение

1.1 Задание

Описать четырьмя графовыми моделями (граф управления, информационный граф, операционная история, информационная история) последовательный алгоритм либо фрагмент алгоритма, содержащий от 15 значащих строк кода и от двух циклов, один из которых является вложенным в другой.

Вариант 17: в качестве реализуемого алгоритма — сортировка слиянием.

1.2 Графовые модели программы

Программа представлена в виде графа: набор вершин и множество соединяющих их направленных дуг.

- 1) **Вершины:** процедуры, циклы, линейный участки, операторы, итерации циклов, срабатывание операторов и т. д.
- 2) **Дуги:** отражают связь (отношение между вершинами).

Выделяют 2 типа отношений:

- 1) операционное отношение — по передаче управления;
- 2) информационное отношение — по передаче данных.

Граф управления — модель, в который **вершины** — операторы, **дуги** — операционные отношения.

Информационный граф — модель, в которой **вершины:** операторы, **дуги** — информационные отношения.

Операционная история — модель, в которой **вершины:** срабатывание операторов, **дуги** — операционные отношения.

Информационная история — модель, в которой **вершины:** срабатывание операторов, **дуги** — информационные отношения.

2 Выполнение

2.1 Выбор языка программирования

Для выполнения домашнего задания был выбран язык C++.

2.2 Код программы

Код программы приведен в листинге 2.1.

Листинг 2.1 – 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){  
    //(3)  
17  
18      for (int left = 0; left < size - 1; left += 2 * step){  
        //(4)  
19  
20          int mid = min(left + step - 1, size - 1),  
              //(5)  
21          right = min(left + 2 * step - 1, size - 1);  
              //(6)  
22
```

```

23     int i = left ,
        //(7)
24     j = mid + 1,
        //(8)
25     pos = 0,
        //(9)
26     len = right - left + 1;
        //(10)
27
28     arrayT tmp(len);
        //(11)
29
30     while (i <= mid || j <= right) {
        //(12)
31
32         if (i <= mid && (j > right || arr[i] < arr[j]))
            { //(13)
33             tmp[pos] = arr[i];
                //(14)
34             i++;
                //(15)
35         }
36         else {
37             tmp[pos] = arr[j];
                //(16)
38             j++;
                //(17)
39         }
40         pos++;
            //(18)
41     }
42
43     for (int k = 0; k < len; ++k)
        //(19)
44         arr[k + left] = tmp[k];
            //(20)
45     }
46 }
47
48     cout << "После: ";
49     for (auto elem : arr)

```

```

50         cout << elem << " ";
51     cout << endl;
52 }

```

2.3 Графовые модели программы

2.3.1 Граф управления

На рисунке 2.1 представлен граф управления.

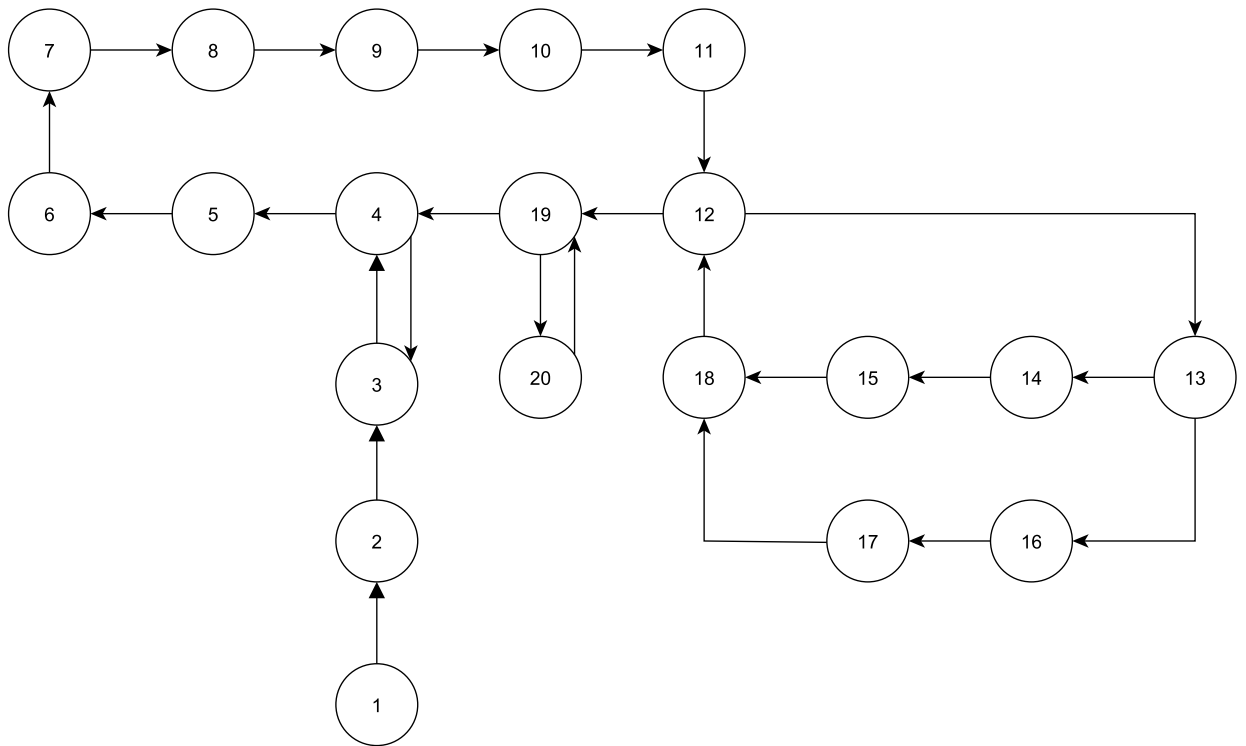


Рисунок 2.1 – Граф управления

2.3.2 Информационный граф

На рисунке 2.2 представлен информационный граф.

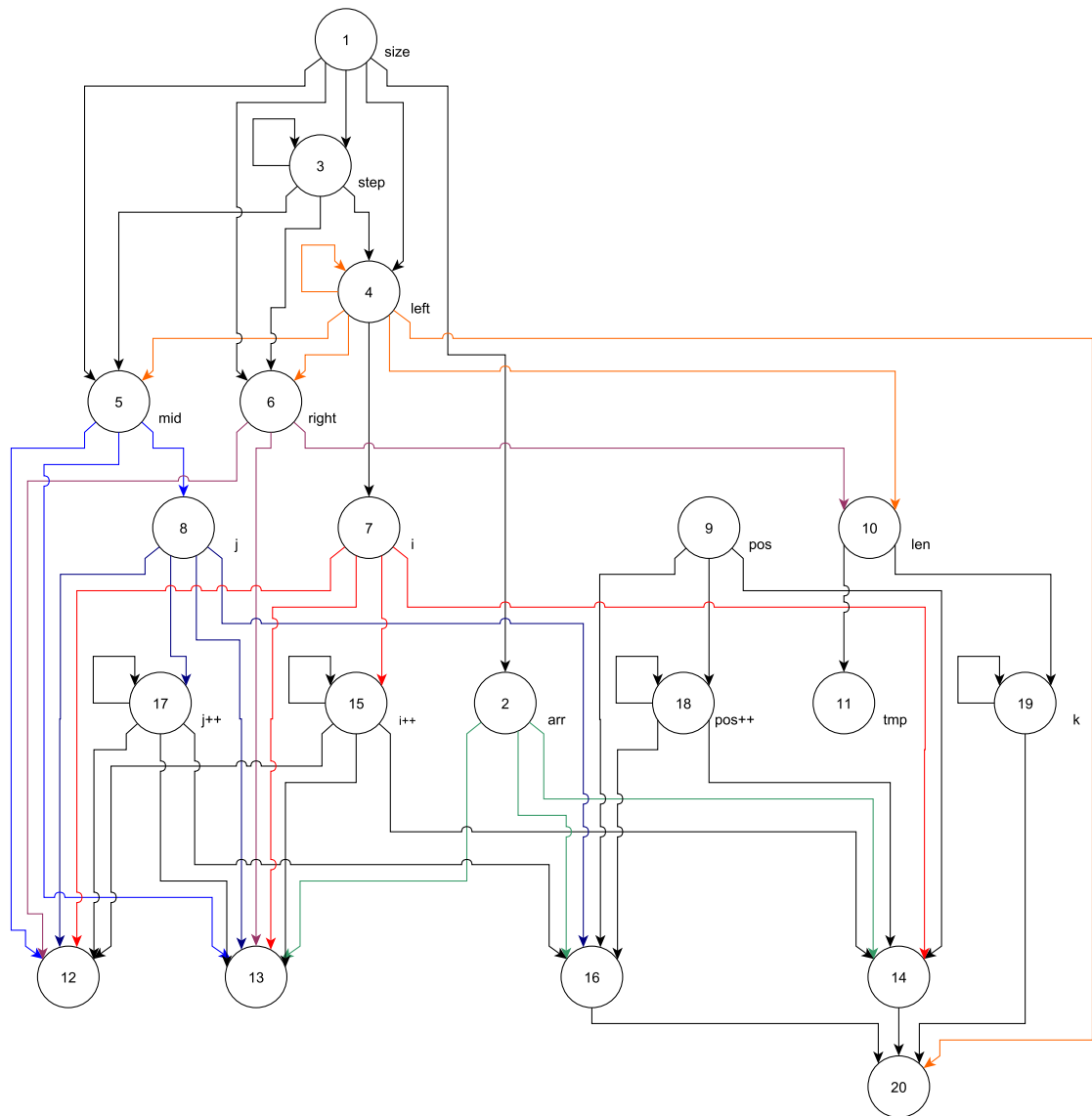


Рисунок 2.2 – Информационный граф

2.3.3 Операционная история

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

На рисунках 2.3–2.4 представлена операционная история программы для этого массива, а также и информационная история 2.5–2.6.

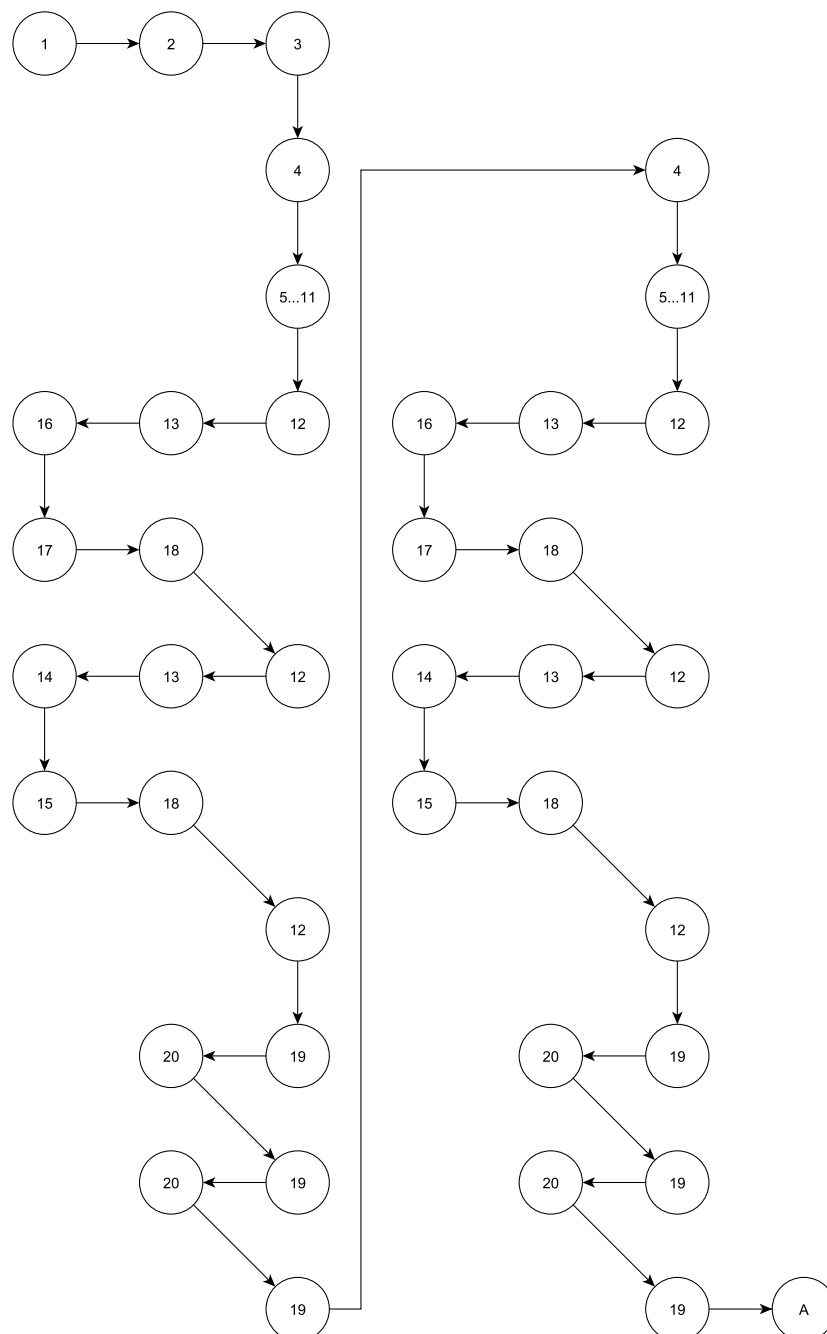


Рисунок 2.3 – Операционная история (начало)

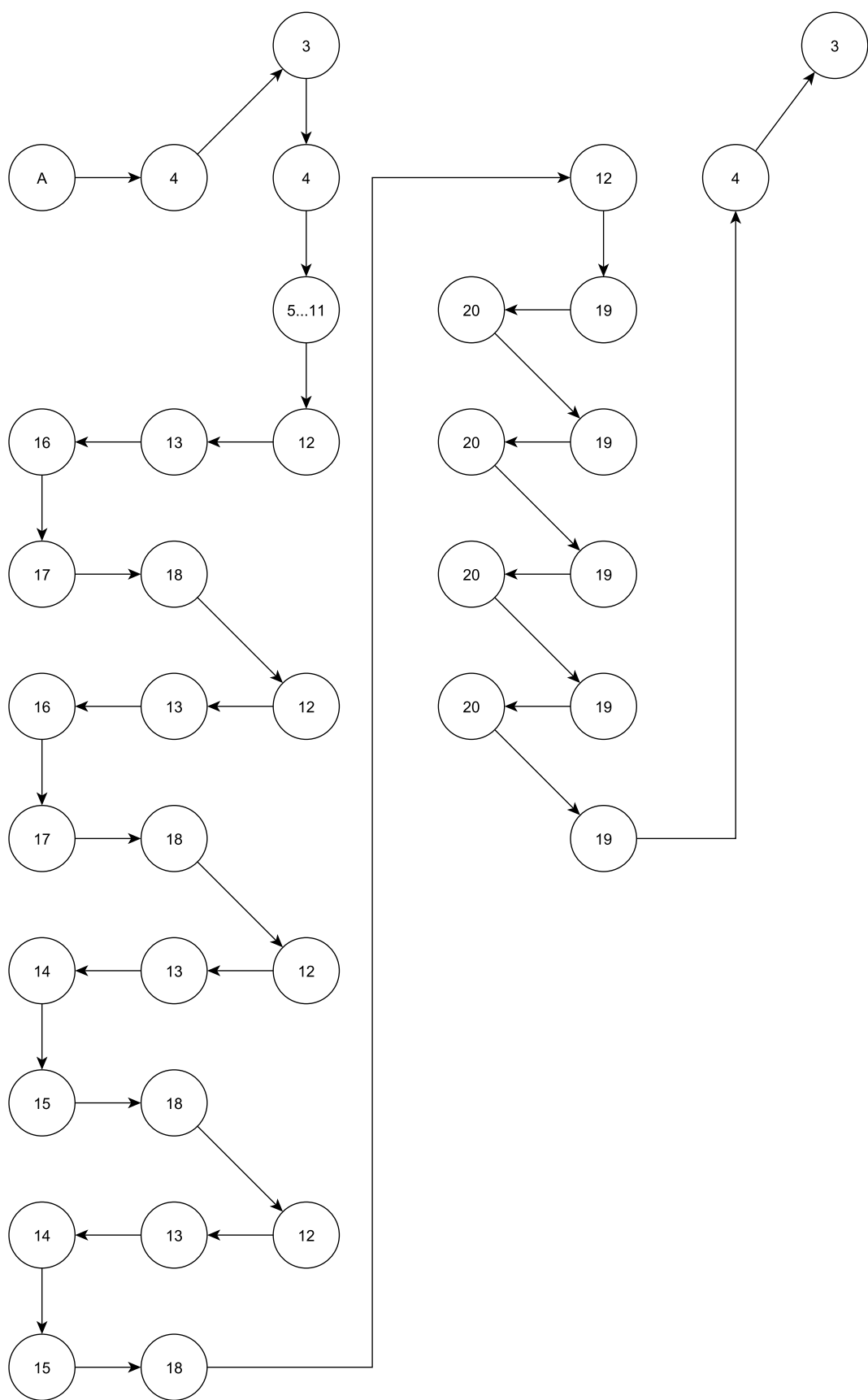


Рисунок 2.4 – Операционная история (конец)

2.3.4 Информационная история

На рисунках 2.5–2.6 представлена информационная история программы для этого массива.

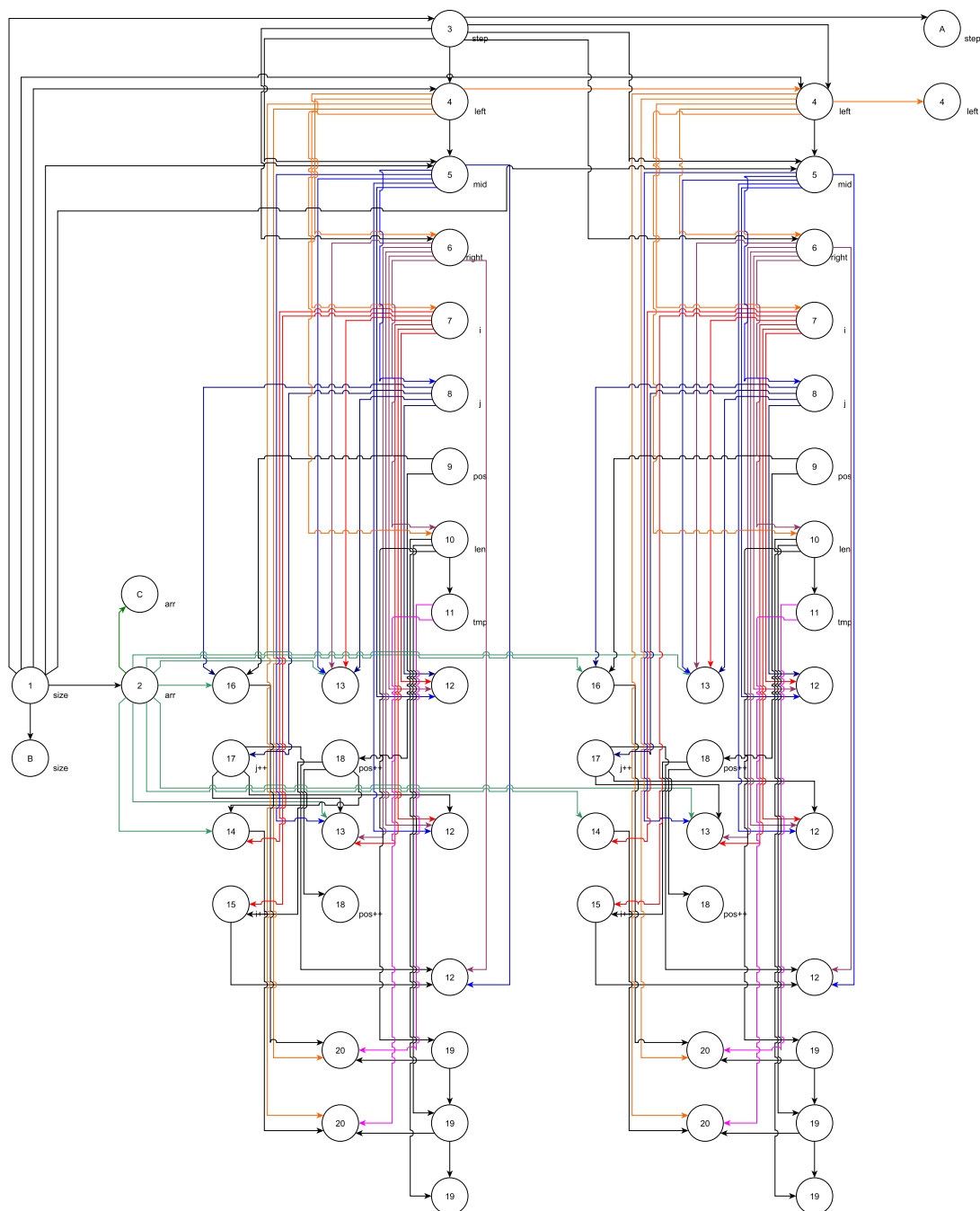


Рисунок 2.5 – Информационная история (начало)

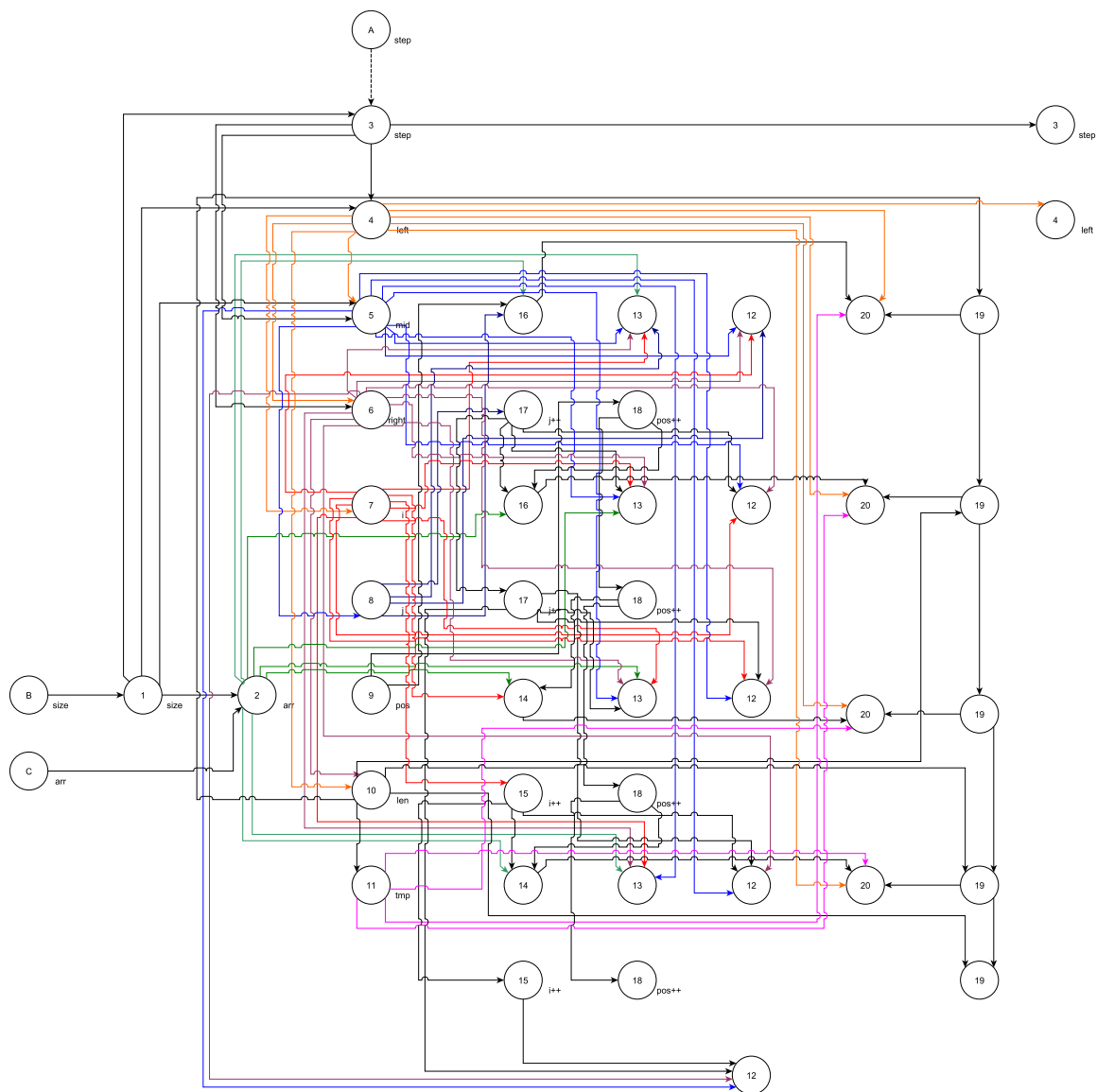


Рисунок 2.6 – Информационная история (конец)

Возможность распараллеливания

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