

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»

Домашнее задание по
дисциплине «Анализ алгоритмов»

Тема Графовые модели программ

Студент Костев Д.И.

Группа ИУ7-51Б

Преподаватели Волкова Л.Л.

Оглавление

1	Исходный код алгоритма	2
2	Модели программ	4
2.1	Граф управления программы	4
2.2	Информационный граф программы	5
2.3	Операционная история программы	6
2.4	Информационная история программы	9

1 | Исходный код алгоритма

Листинг 1.1: Функция побитовой сортировки

```
1 void RadixSortStep(TNode* source, TNode* dest, unsigned int n, unsigned int*  
  offset, unsigned char sortable_bit)  
2 {  
3     unsigned char* b = (unsigned char*)&source[n].key + sortable_bit; (12.1)  
4     TNode* v = &source[n]; (12.2)  
5     while (v >= source) (12.3)  
6     {  
7         dest[--offset[*b]] = *v--; (12.4)  
8         b -= sizeof(TNode); (12.5)  
9     }  
10 }  
11  
12 void RadixSort(TNode* m, unsigned int n) (0)  
13 {  
14     TNode* m_temp = (TNode*)malloc(sizeof(TNode) * n); (1)  
15  
16     unsigned int s[sizeof(m->key) * 256] = { 0 }; (2)  
17     unsigned char* b = (unsigned char*)&m[n - 1].key; (3)  
18     while (b >= (unsigned char*)&m[0].key) (4)  
19     {  
20         for (unsigned int digit = 0; digit < sizeof(m->key); digit++) (5)  
21             s[(b + digit) + 256 * digit]++; (6)  
22         b -= sizeof(TNode); (7)  
23     }  
24  
25     for (unsigned int i = 1; i < 256; i++) (8)  
26         for (unsigned int digit = 0; digit < sizeof(m->key); digit++) (9)  
27             s[i + 256 * digit] += s[i - 1 + 256 * digit]; (10)  
28  
29     for (unsigned int digit = 0; digit < sizeof(m->key); digit++) (11)  
30     {  
31         RSort_step(m, m_temp, n - 1, &s[256 * digit], digit); (12)  
32         TNode* temp = m; (13)  
33         m = m_temp; (14)  
34         m_temp = temp; (15)  
35     }  
36  
37     if (sizeof(m->key) == 1) (16)
```

38	{	
39	TNode* temp = m;	(17)
40	m = m_temp;	(18)
41	m_temp = temp;	(19)
42	memcpy(m, m_temp, n * sizeof(TNode));	(20)
43	}	
44		
45	free(m_temp);	(21)
46	}	

2 | Модели программ

2.1 Граф управления программы

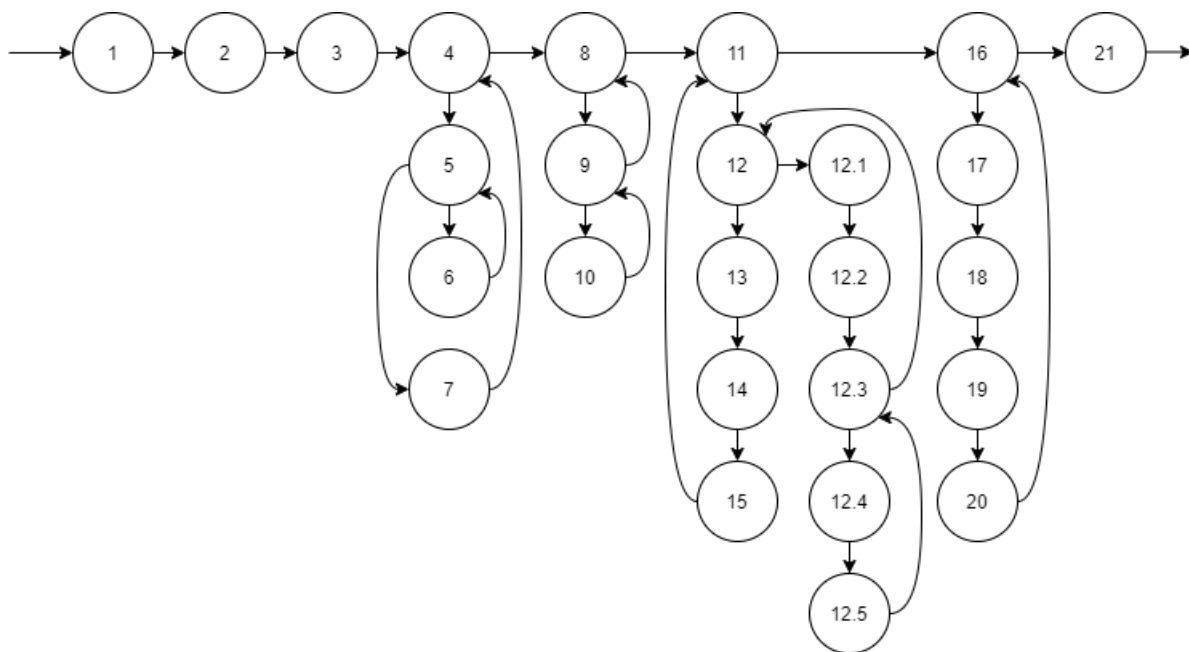


Рис. 2.1: Граф управления программы

2.2 Информационный граф программы

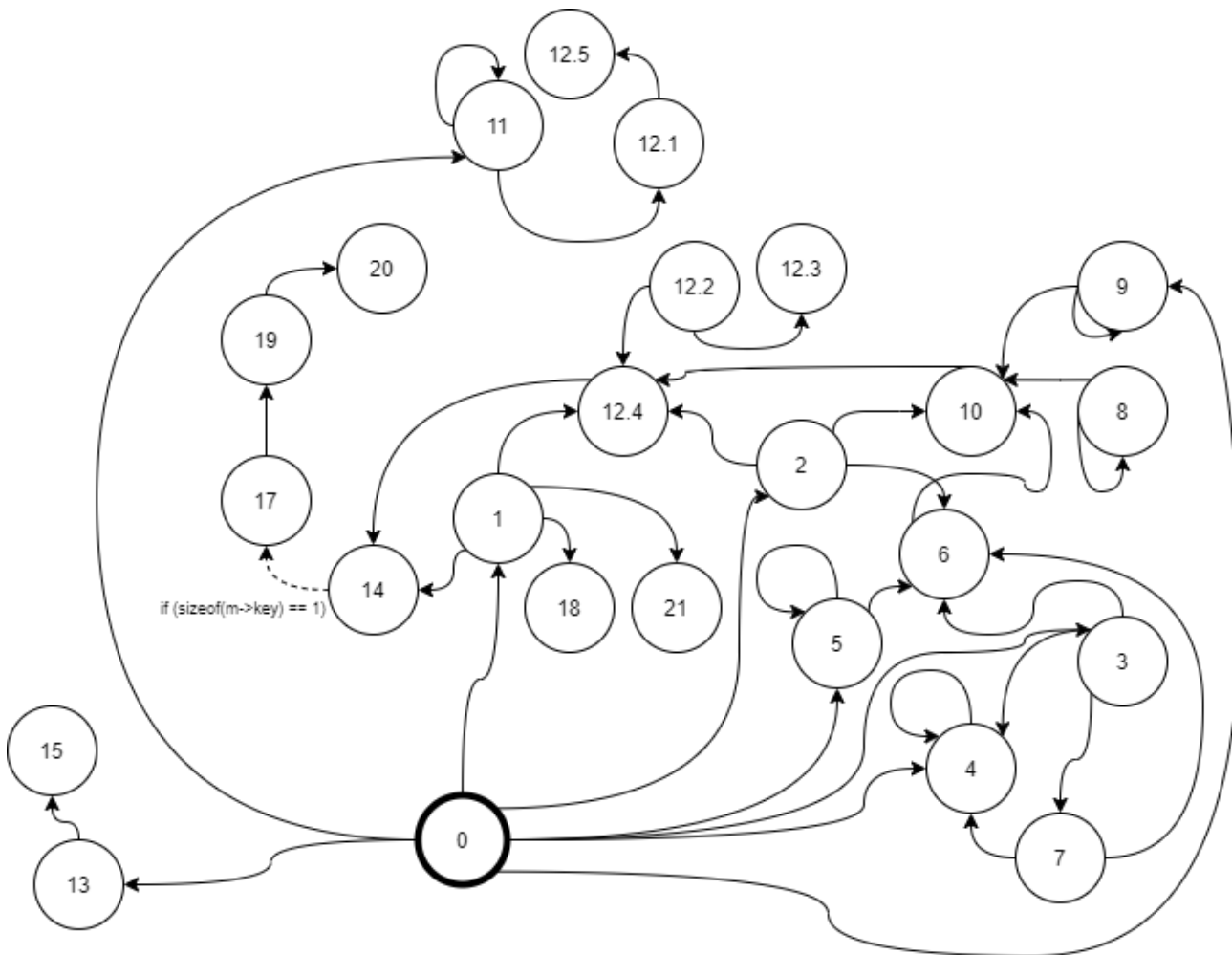


Рис. 2.2: Информационный граф программы

2.3 Операционная история программы

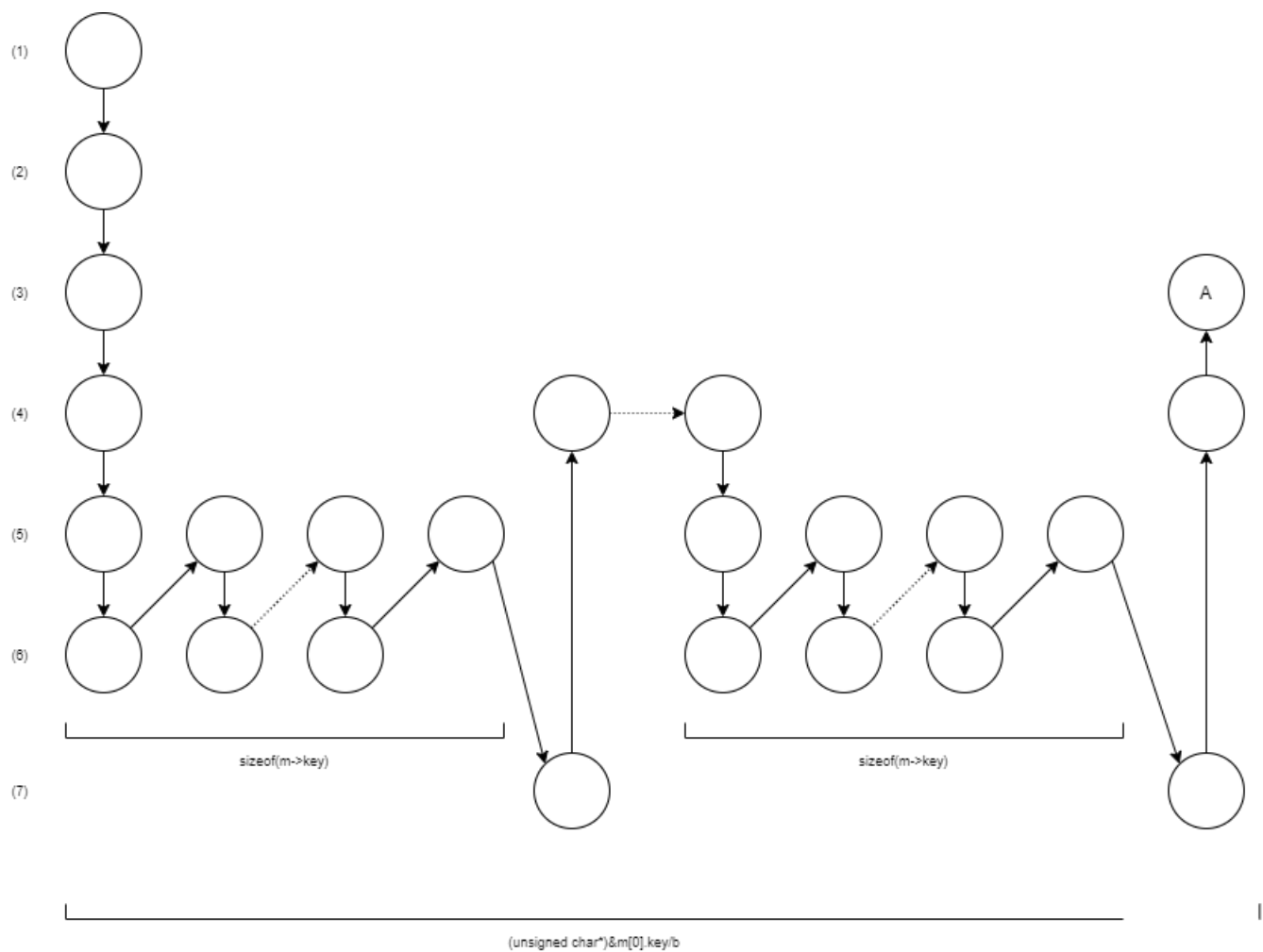


Рис. 2.3: Операционная история программы, часть 1

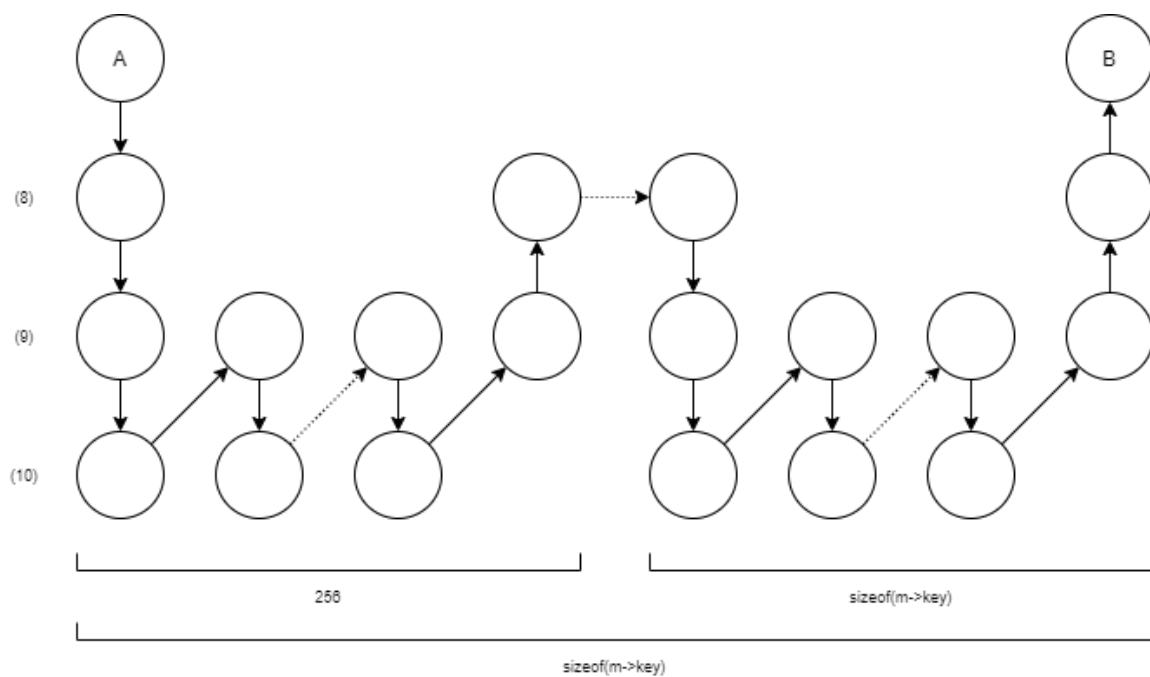


Рис. 2.4: Операционная история программы, часть 2

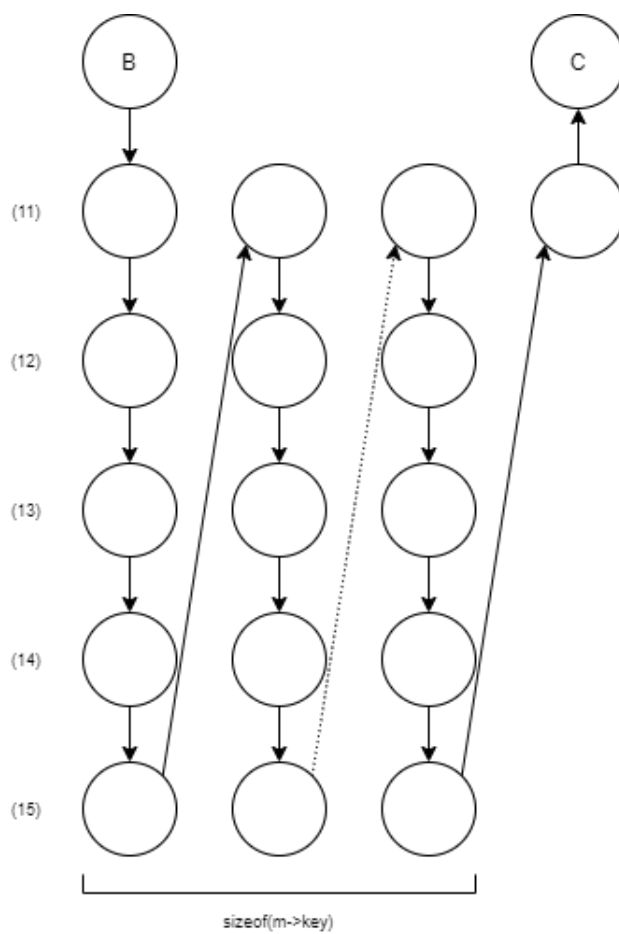


Рис. 2.5: Операционная история программы, часть 3

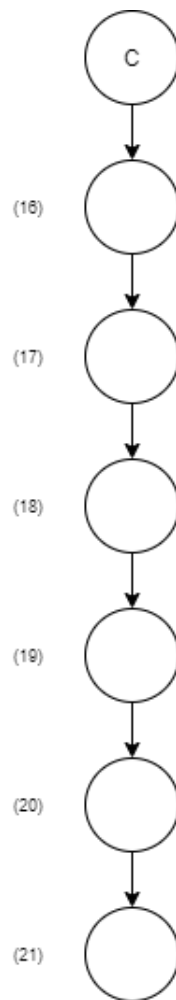


Рис. 2.6: Операционная история программы, часть 4

2.4 Информационная история программы

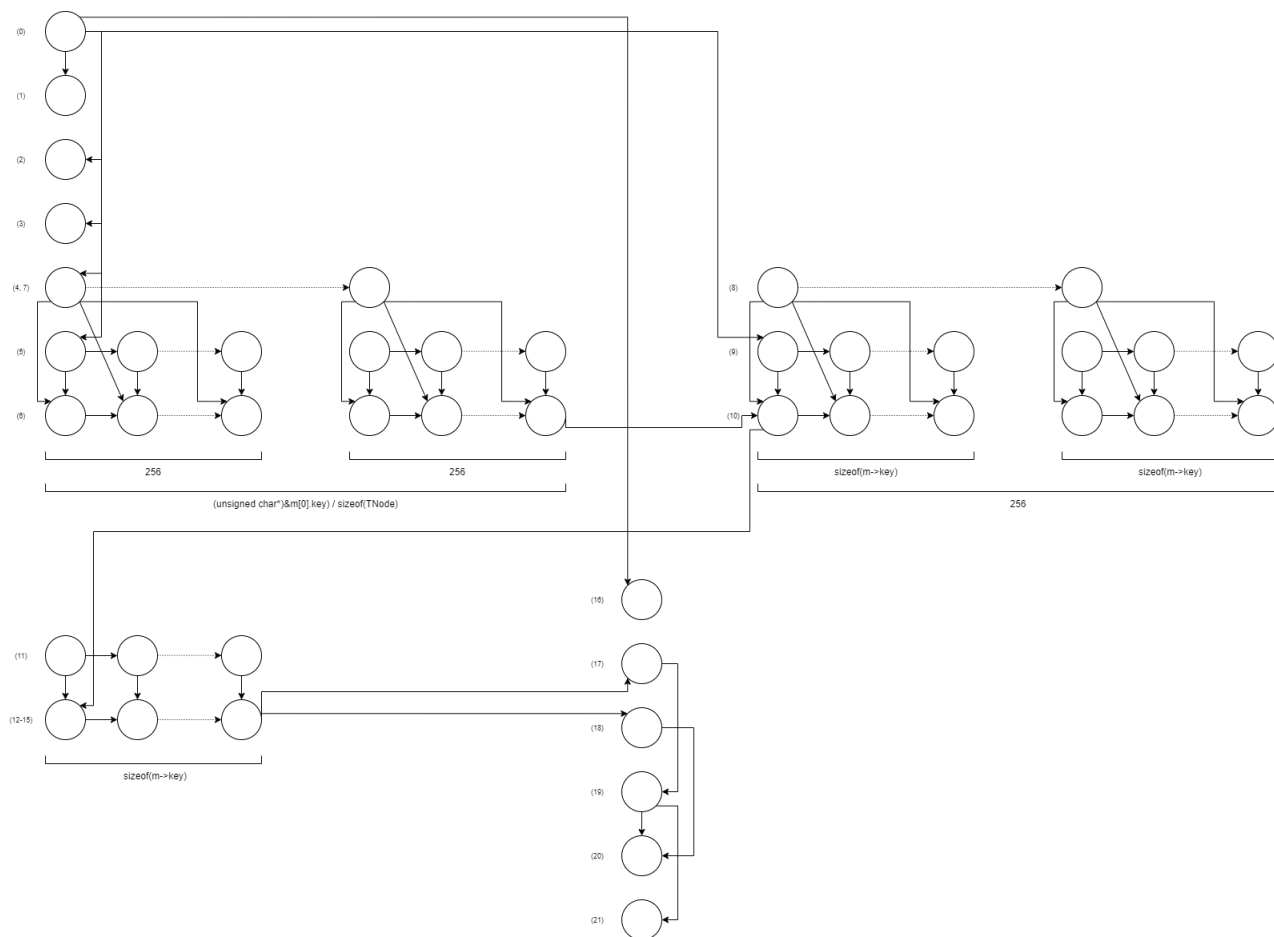


Рис. 2.7: Информационная история программы.