

## Алгоритм сбалансированного многопутевого слияния

Алгоритм сбалансированного многопутевого слияния основан на алгоритме естественного слияния и имеет следующие особенности:

- 1) Для сортировки используется  $N$  файлов, доступных для хранения промежуточных результатов (значение  $N$  предполагается четным)
- 2) Алгоритм имеет одну фазу
- 3) Серии распределяются по первым  $N/2$  файлам, далее производится их слияние на вторые  $N/2$  файлов

### Входная последовательность:

8	9	1	5	7	8	3	4	5	1	9	2	5	4	6	8	3	1	7	4	5	2	3	8	7	9	3	4	5
1	2	3	4	5	6	7	8	9	10	11	12																	

### Распределение по сериям

#### Проход 1

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

- 1: 

1	4	7	10
---	---	---	----
- 2: 

2	5	8	11
---	---	---	----
- 3: 

3	6	9	12
---	---	---	----
- 4:
- 5:
- 6:

#### Проход 2

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1:	1	4	7	10
2:	2	5	8	11
3:	3	6	9	12
4:	1, 2, 3			10, 11, 12
5:	4, 5, 6			
6:	7, 8, 9			

Проход 2

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1:	1, 2, 3, 4, 5, 6, 7, 8, 9										
2:	10, 11, 12										
3:											
4:	1, 2, 3					10, 11, 12					
5:	4, 5, 6										
6:	7, 8, 9										

Проход 3

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1: 1, 2, 3, 4, 5, 6, 7, 8, 9

2: 10, 11, 12

3:

4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

5:

6:

Анализ алгоритма сбалансированного многопутевого слияния

- Пусть во входном файле имеется  $r$  серий, для сортировки используется  $N$  файлов,  $N_h = N/2$
- Серии поровну распределяются по  $N_h$  файлам. Следовательно, в каждом из них образуется  $r/N_h$  серий
- При слиянии файлов первые серии в каждом из них объединяются в одну, тоже самое происходит со вторыми, третьими, ...  $N_h$ -ми сериями. В итоге количество серий будет сокращено с  $r$  до  $r/N_h$
- Таким образом, каждый проход уменьшает число серий в  $N_h$  раз
- Число проходов не превышает  $\log(N_h * n)$

Реализация алгоритма

■ Файл представляется в программе через структуру `struct file`, описанную ранее.

■ Для управления набором из  $N$  файлов необходимо использовать массив: `struct file tapes[N];`

■  $N$ -файловый набор  $F$  разбивается пополам ( $F1$  и  $F2$ ):

■  $F1$  содержит промежуточный результат (серии равномерно распределены по файлам)

■ Серии из  $F1$  сливаются в  $F2$ .

■ Переключение ролей:  $F1 \leftrightarrow F2$ .

Отображение файлов

Отображение реализуется с помощью целочисленного массива:

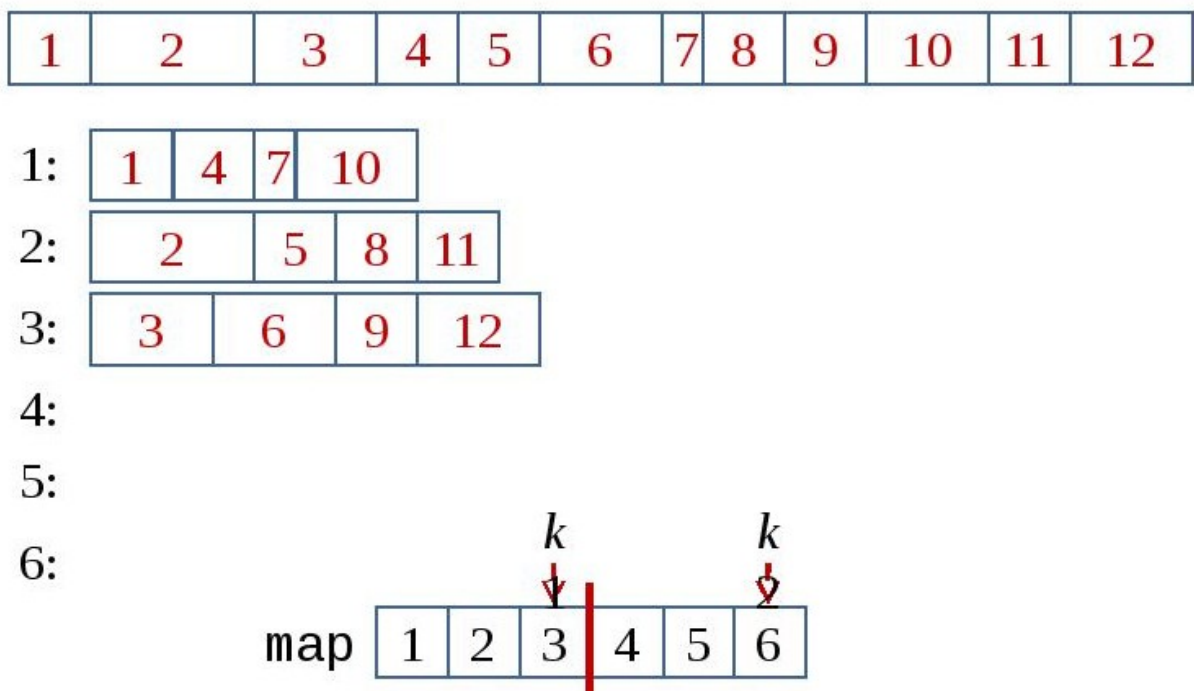
```
int map[N]; // (N % 2) == 0
```

Массив map содержит **индексы** элементов массива tapes (struct file tapes[N]).

Первая половина map: map[0]...map[N/2-1] содержит индексы файлов, **из которых** производится слияние.

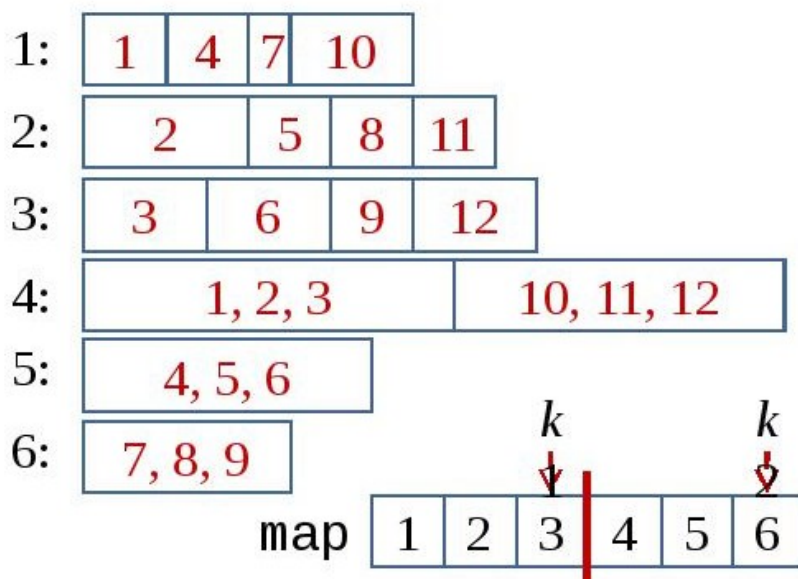
Вторая половина map: map[N/2]...map[N-1] хранит индексы файлов **в которые** производится.

Анализ алгоритма (проход 1)



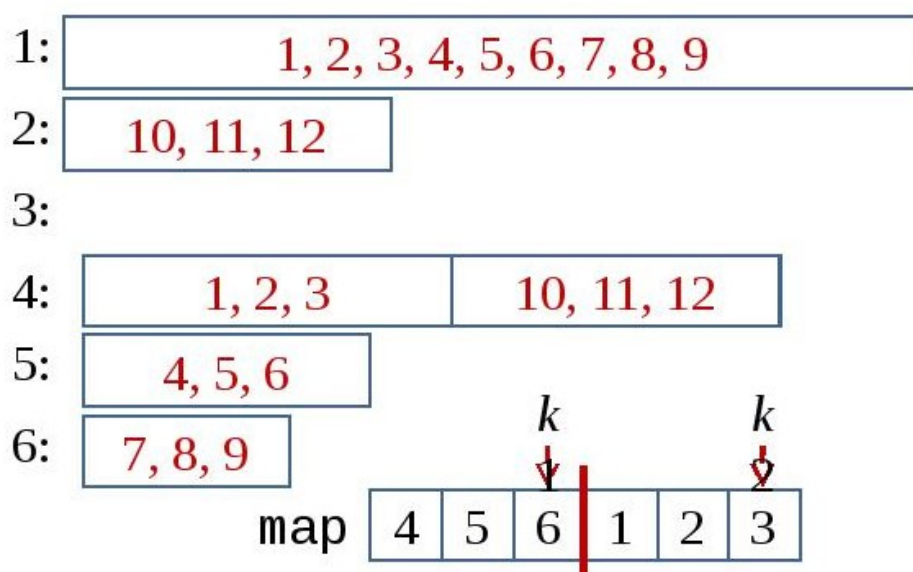
Анализ алгоритма (проход 2)

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----



Анализ алгоритма (проход 2)

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----



Анализ алгоритма (проход 3)

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1: 1, 2, 3, 4, 5, 6, 7, 8, 9

2: 10, 11, 12

3:

4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

5:

6: map

1	2	3		4	5	6
---	---	---	--	---	---	---

$k$  ↓                      ↓  $k$   
 $\downarrow$                        $\downarrow$

Анализ алгоритма (проход 3)

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1:

2:

3:

4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

5:

6: map

4	5	6		1	2	6
---	---	---	--	---	---	---

$k$  ↓                      ↓  $k$   
 $\downarrow$                        $\downarrow$

Процедура слияния серий из N файлов



1.  $k$  - номер выходного файла (изначально  $k = N/2$ )
2. Проверяется пустота каждого файла  $tapes[map[i]]$ ,  $i = 0 \dots k_1-1$ . Если  $i$ -й файл пуст, то:  
 $map[i] = map[k_1-1]$ ,  $k_1=k_1-1$ .
3. Определяется наименьший элемент среди входных файлов  $v=\min(tapes[map[i]] , i = 0 \dots k_1)$
4.  $write(v, tapes[map[k]])$
5.  $k = k+1$ , если  $k \geq N$   $k = N/2$