

# Алгоритмы и структуры данных-1

## Бинарная куча и внешняя сортировка

Практическое занятие 8 — 21.10–02.11.2024  
2024–2025 учебный год

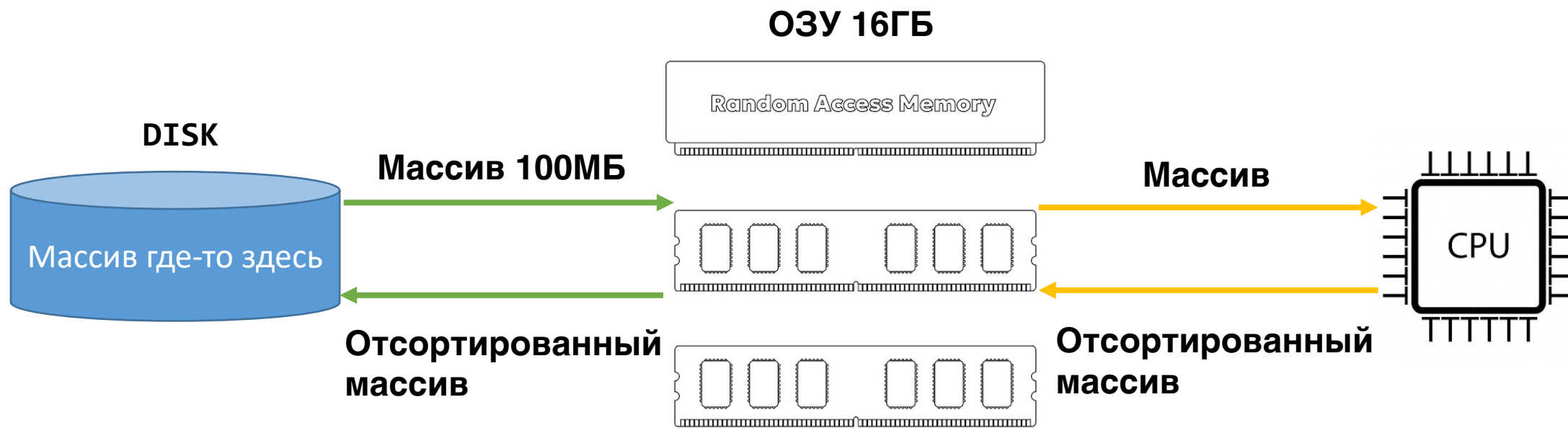
# Операции с бинарной кучей

# Создание бинарной max-кучи

1. Какую структуру будет иметь бинарная max-куча, созданная на основании последовательности следующих элементов:  
[19, 34, 23, 16, 54, 89, 24, 29, 15, 61, 27]
2. Какую структуру будет иметь полученная max-куча после выполнения двух шагов алгоритма HEAPSORT?

# Внутренняя и внешняя сортировка

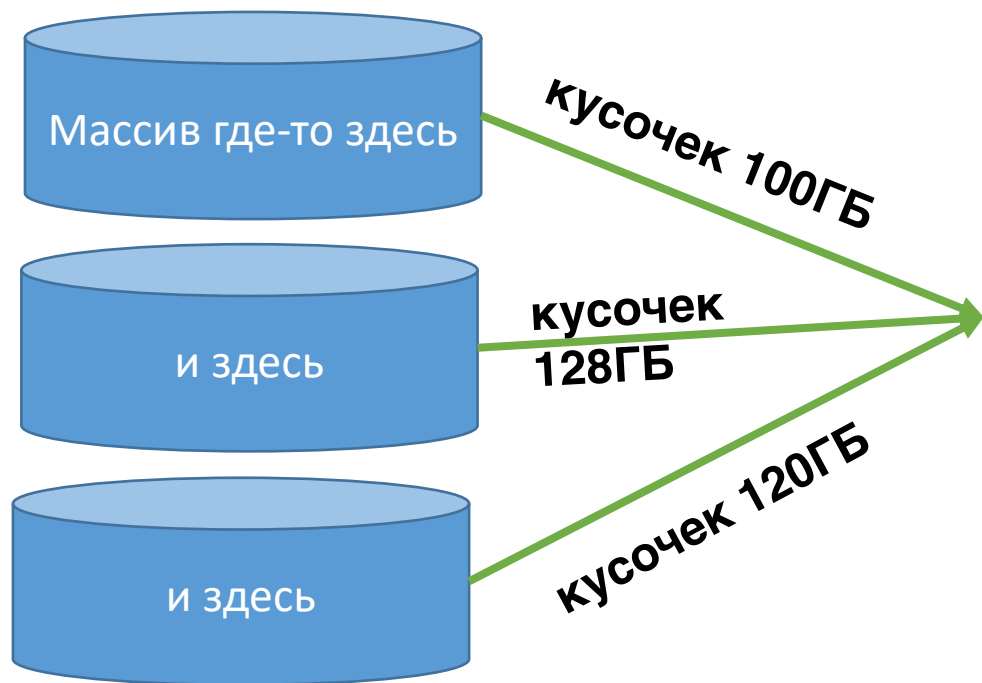
# Внутренняя сортировка



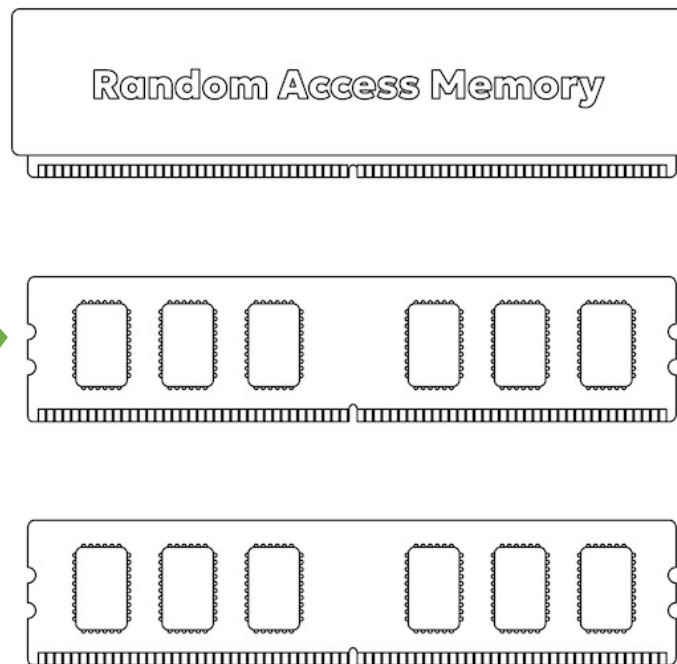
Как решить задачу?

# Внешняя сортировка

Хранилище на петабайт



128ГБ ОЗУ



Массив не помещается в ОЗУ, что делать?



# Внешний MERGE SORT

Наивный взгляд

# Наивный подход

Шаг 1 Разделяем входной файл **input** на два **input1** и **input2**

<b>input</b>	<b>36</b>	<b>17</b>	<b>28</b>	<b>23</b>	<b>20</b>	<b>13</b>	<b>15</b>	<b>14</b>
--------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------



# Наивный подход

Шаг 1 Разделяем входной файл **input** на два **input1** и **input2**, а также загружаем по одному блоку из каждого файла в буфер

input1	36	17	28	23
--------	----	----	----	----

input2	20	13	15	14
--------	----	----	----	----

# Наивный подход

Шаг 2 Запускаем MERGE SORT, выбирая по одной записи из каждого входного буфера

input1 

36	17	28	23
----	----	----	----

input2 

20	13	15	14
----	----	----	----

output1 

--	--	--	--

output2 

--	--	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1		17	28	23
--------	--	----	----	----

input2		13	15	14
--------	--	----	----	----

output1	20	36		
---------	----	----	--	--

output2				
---------	--	--	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1		17	28	23
--------	--	----	----	----

input2		13	15	14
--------	--	----	----	----

output1	20	36		
---------	----	----	--	--

output2				
---------	--	--	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1			28	23
--------	--	--	----	----

input2			15	14
--------	--	--	----	----

output1	20	36		
---------	----	----	--	--

output2	13	17		
---------	----	----	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1			28	23
--------	--	--	----	----

input2			15	14
--------	--	--	----	----

output1	20	36		
---------	----	----	--	--

output2	13	17		
---------	----	----	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1				23
--------	--	--	--	----

input2				14
--------	--	--	--	----

output1	20	36	15	28
---------	----	----	----	----

output2	13	17		
---------	----	----	--	--

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1				23
--------	--	--	--	----

input2				14
--------	--	--	--	----

output1	20	36	15	28
---------	----	----	----	----

output2	13	17		
---------	----	----	--	--



# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1 

--	--	--	--

input2 

--	--	--	--

output1 

20	36	15	28
----	----	----	----

output2 

13	17	14	23
----	----	----	----

# Наивный подход

Шаг 2 Выполняем слияние выбранных записей и записываем результат, поочерёдно меняя буферы вывода

input1

input2

1. Если данные в одном из входных буферов закончились, то необходимо загрузить новую порцию из файла
2. Если исчерпалась емкость буферов вывода, то необходимо выгрузить информацию на диск

36

15

28

17

14

23

# Наивный подход

Шаг 3 Загружаем информацию из выходных файлов в буферы ввода

input1 

--	--	--	--

input2 

--	--	--	--

output1 

20	36	15	28
----	----	----	----

output2 

13	17	14	23
----	----	----	----

# Наивный подход

Шаг 3 Загружаем информацию из выходных файлов в буферы ввода

input1 

20	36	15	28
----	----	----	----

input2 

13	17	14	23
----	----	----	----

output1 

--	--	--	--

output2 

--	--	--	--

# Наивный подход

Шаг 4 Запускаем MERGE SORT, выбирая по две записи из каждого входного буфера

input1	20	36	15	28
--------	----	----	----	----

input2	13	17	14	23
--------	----	----	----	----

output1				
---------	--	--	--	--

output2				
---------	--	--	--	--

# Наивный подход

Шаг 4 Выполняем слияние выбранных пар записей и записываем результат, поочередно меняя буферы вывода

input1			15	28
--------	--	--	----	----

input2			14	23
--------	--	--	----	----

output1	13	17	20	36
---------	----	----	----	----

output2				
---------	--	--	--	--

# Наивный подход

Шаг 4 Выполняем слияние выбранных пар записей и записываем результат, поочередно меняя буферы вывода

input1			15	28
--------	--	--	----	----

input2			14	23
--------	--	--	----	----

output1	13	17	20	36
---------	----	----	----	----

output2				
---------	--	--	--	--

# Наивный подход

Шаг 4 Выполняем слияние выбранных пар записей и записываем результат, поочередно меняя буферы вывода

input1 

--	--	--	--

input2 

--	--	--	--

output1 

13	17	20	36
----	----	----	----

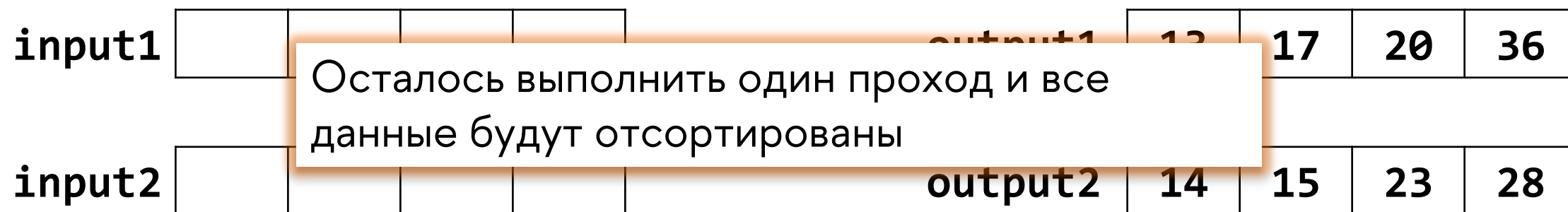
output2 

14	15	23	28
----	----	----	----



# Наивный подход

Шаг 4 Выполняем слияние выбранных пар записей и записываем результат, поочередно меняя буферы вывода



# Наивный подход

Использует достоинства *двойной буферизации* – буфер ввода и вывода меняется местами

# Наивный подход

Использует достоинства *двойной буферизации* – буфер ввода и вывода меняется местами

Для сортировки файла с  $n$  записями потребуется  $n \log n$  проходов, которые связаны с чтением и записи с диска

# Наивный подход

Использует достоинства *двойной буферизации* – буфер ввода и вывода меняется местами

Для сортировки файла с  $n$  записями потребуется  $n \log n$  проходов, которые связаны с чтением и записи с диска

Количество проходов можно уменьшить, если не применять MERGE SORT на первых шагах для малых размеров блоков записей

# Наивный подход

Использует достоинства *двойной буферизации* – буфер ввода и вывода меняется местами

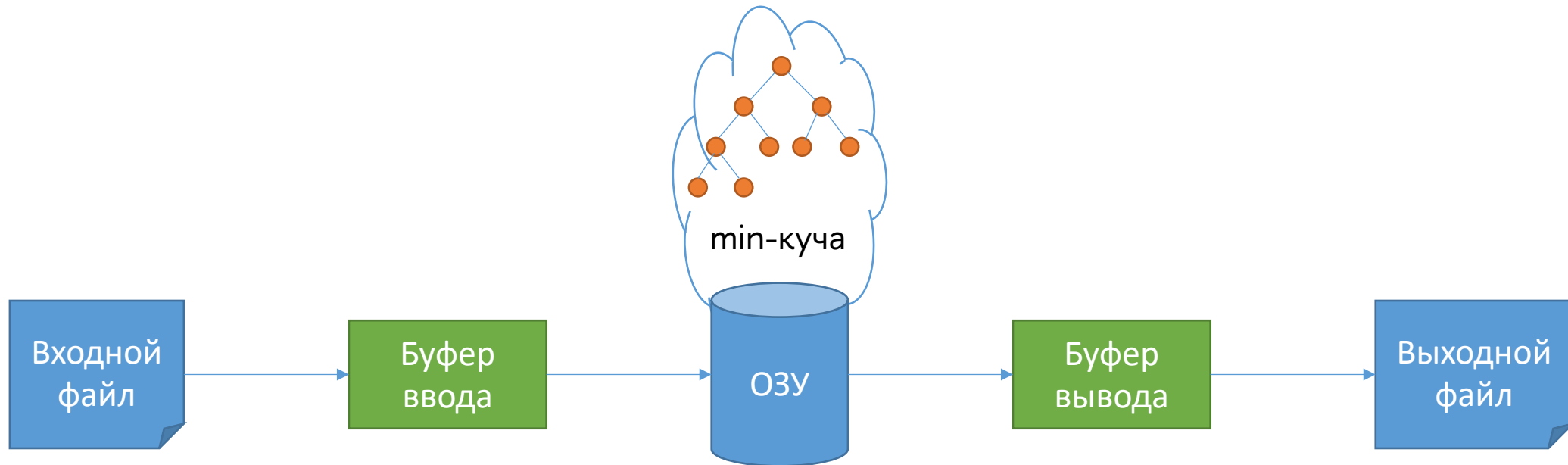
Для сортировки файла с  $n$  записями потребуется  $n \log n$  проходов, которые связаны с чтением и записи с диска

Задача в том, чтобы из входного файла  
создать блоки отсортированных данных  
для слияния **как можно большего размера**

# Внешний MERGE SORT

REPLACEMENT SELECTION и K-WAY MERGE

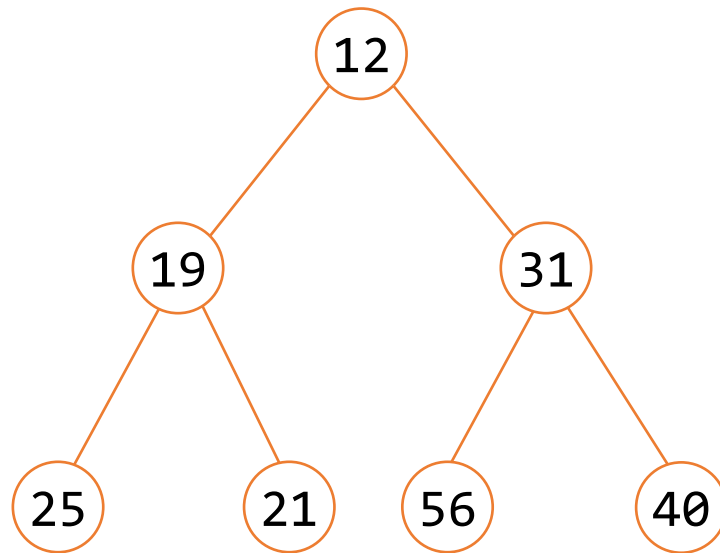
# Общая схема



# Пример работы

input

16	29	14	35	23
----	----	----	----	----



**min-куча**

output

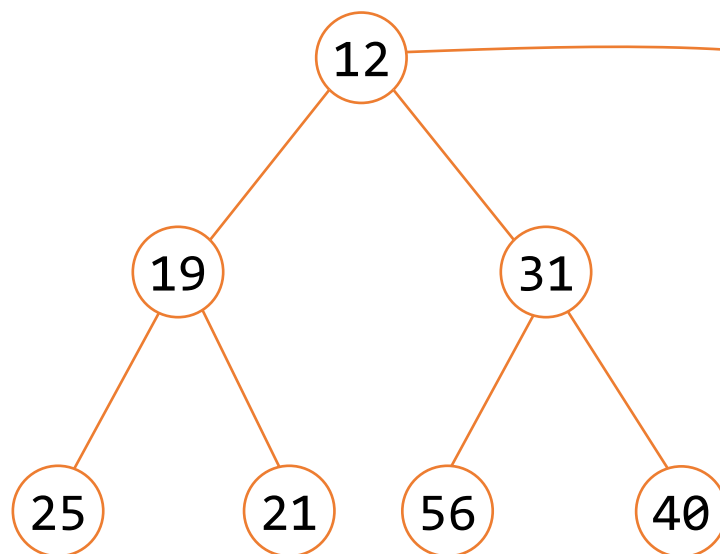
--	--	--	--	--



# Пример работы

input

16	29	14	35	23
----	----	----	----	----



**min-куча**

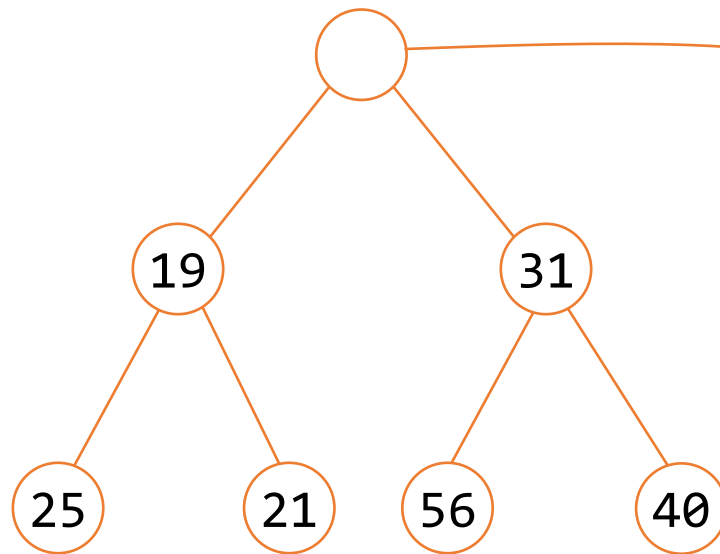
output

--	--	--	--	--

# Пример работы

input

16	29	14	35	23
----	----	----	----	----

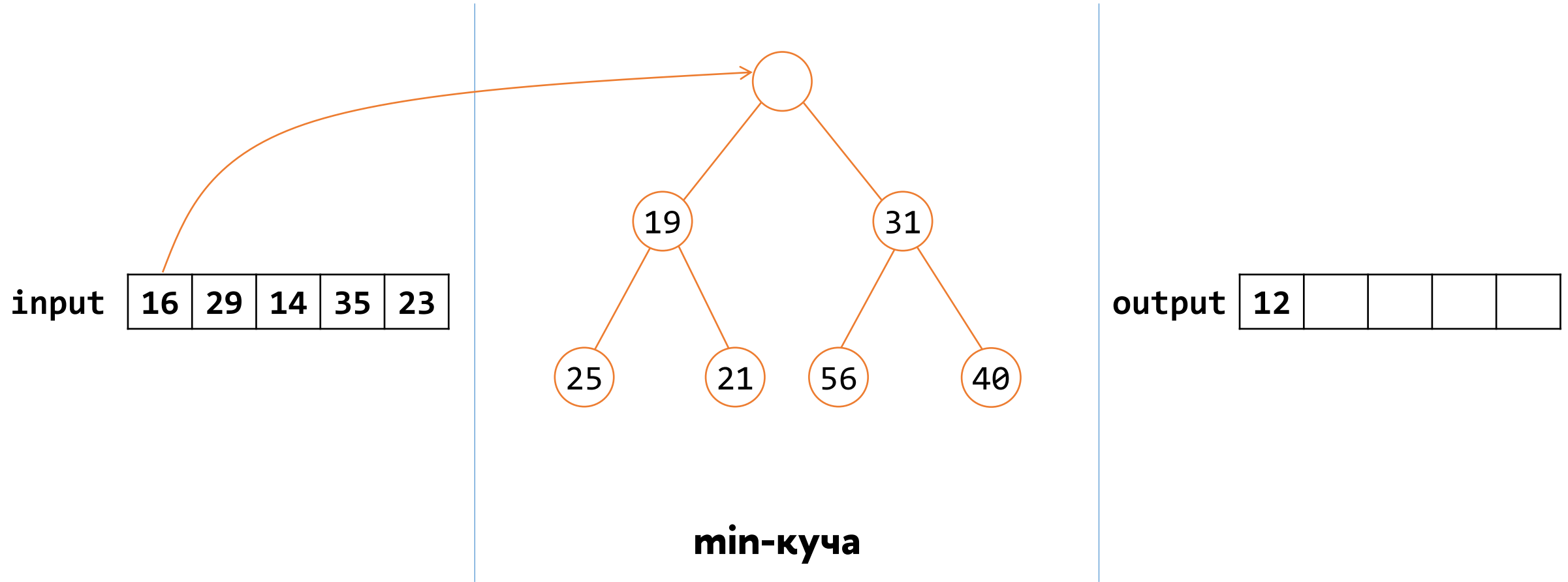


**min-куча**

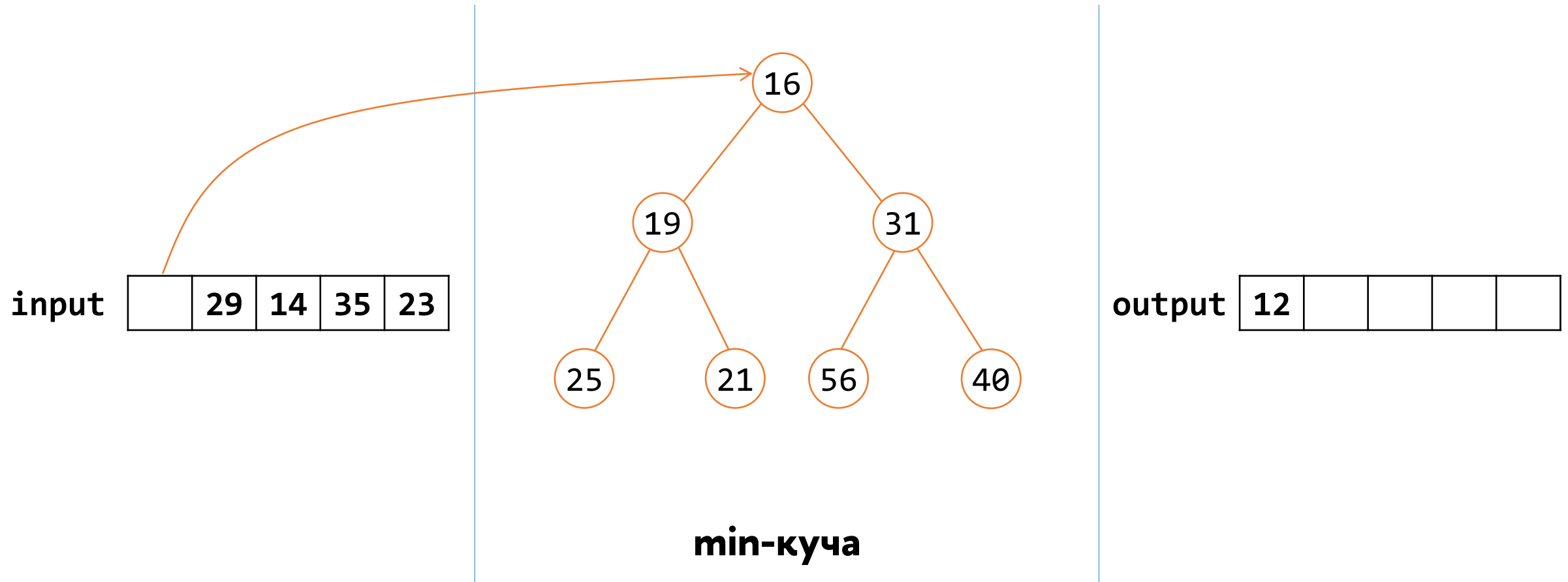
output

12				
----	--	--	--	--

# Пример работы



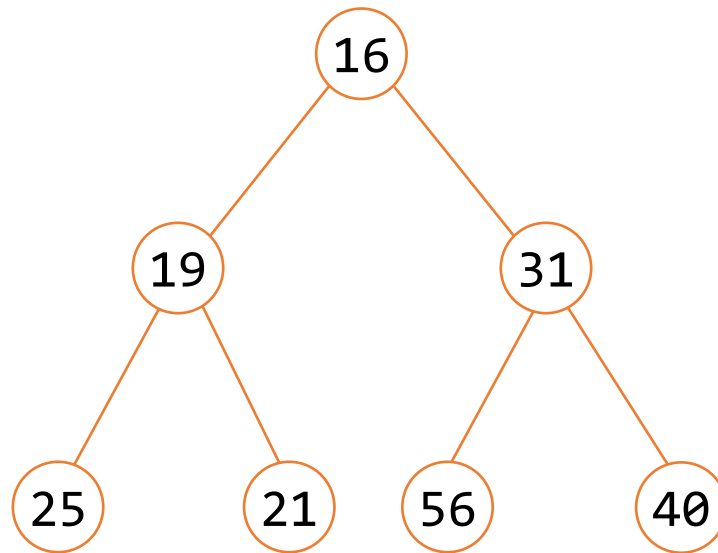
# Пример работы



# Пример работы

input

	29	14	35	23
--	----	----	----	----



**min-куча**

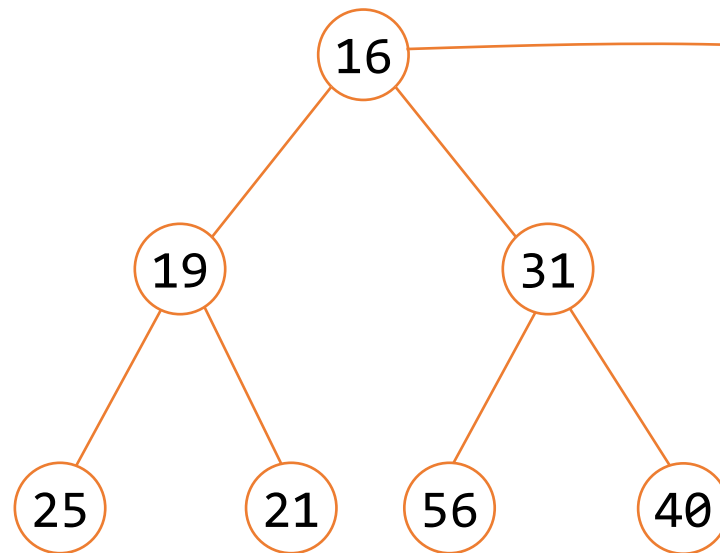
output

12				
----	--	--	--	--

# Пример работы

input

	29	14	35	23
--	----	----	----	----



**min-куча**

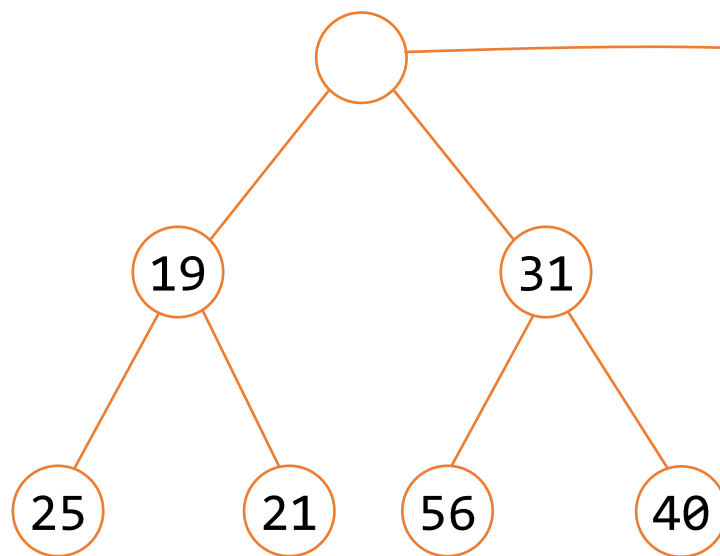
output

12				
----	--	--	--	--

# Пример работы

input

	29	14	35	23
--	----	----	----	----



**min-куча**

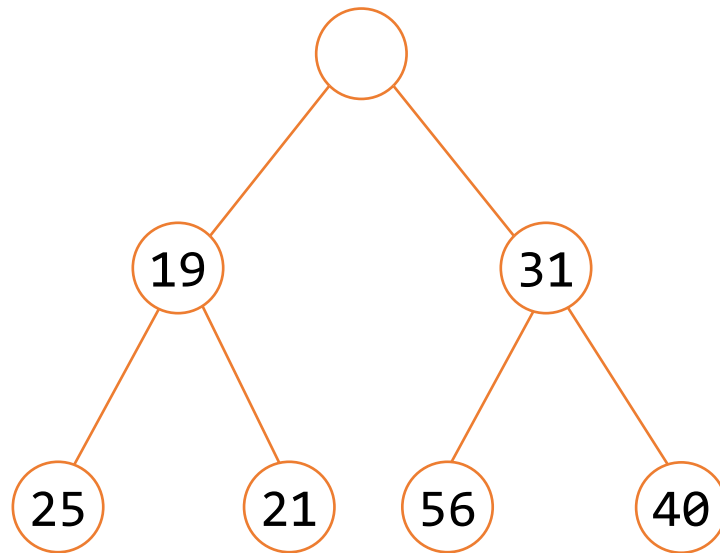
output

12	16			
----	----	--	--	--

# Пример работы

input

	29	14	35	23
--	----	----	----	----



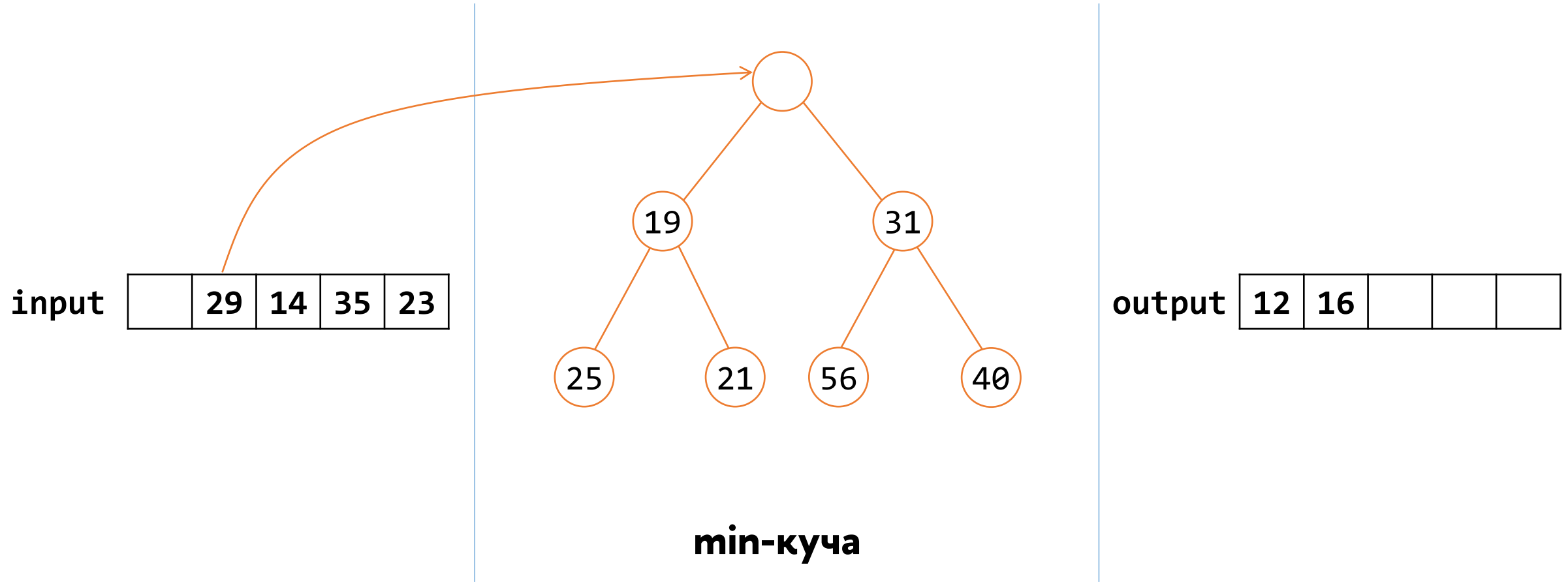
**min-куча**

output

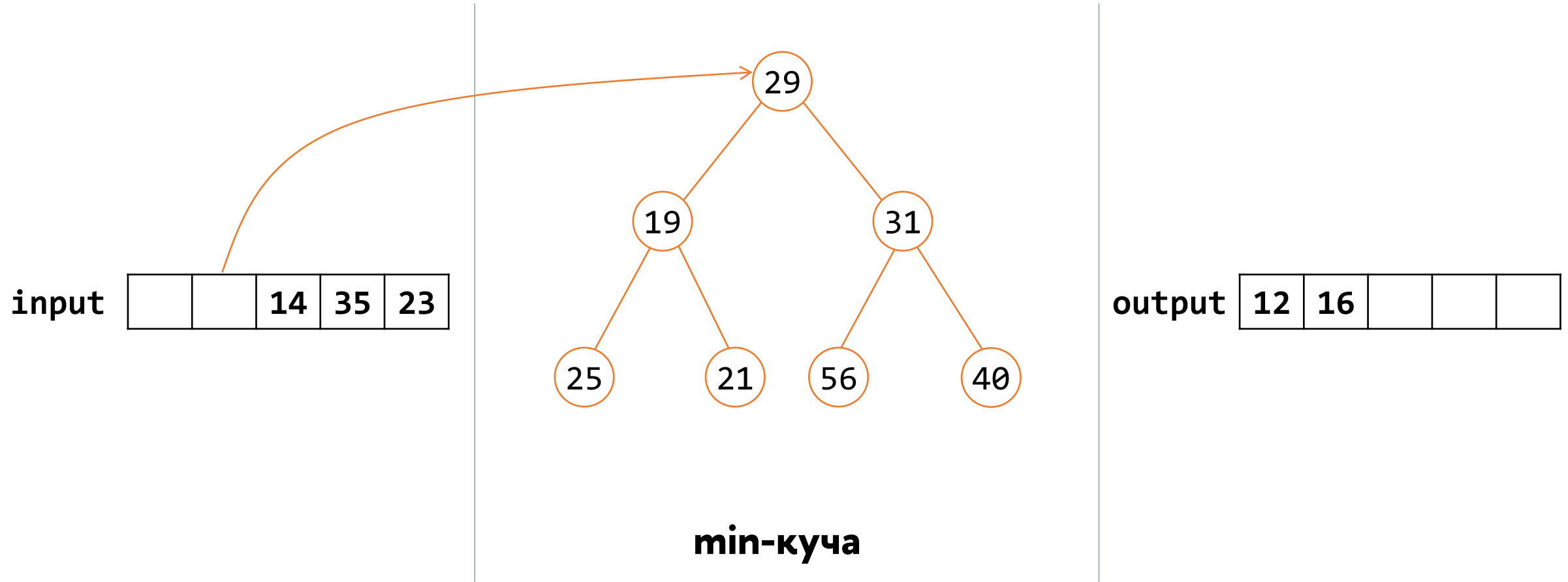
12	16			
----	----	--	--	--



# Пример работы



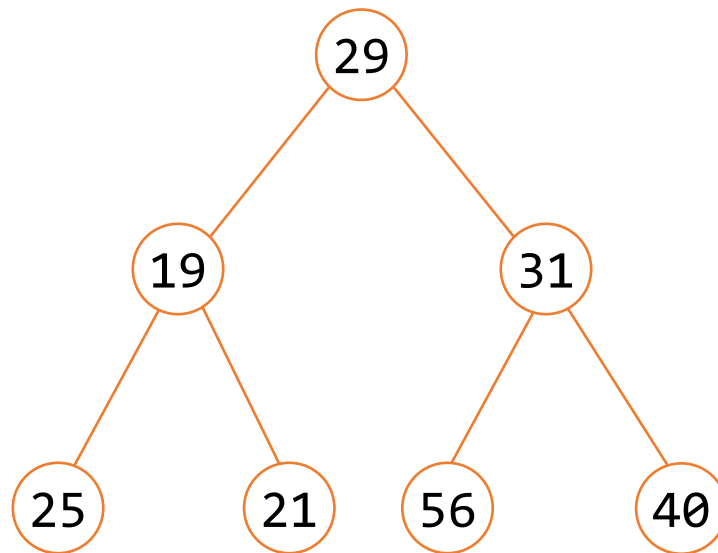
# Пример работы



# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

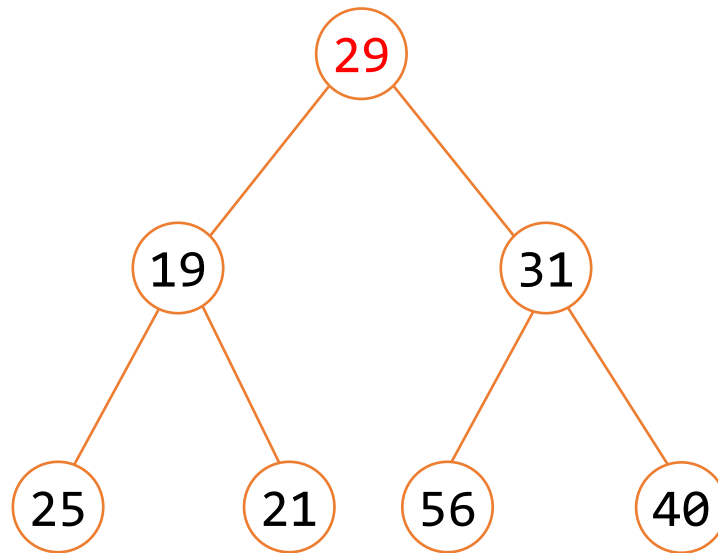
output

12	16			
----	----	--	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

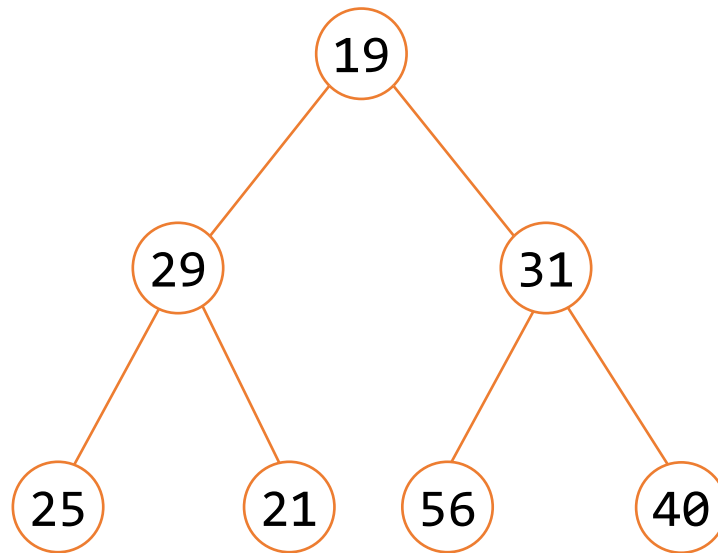
output

12	16			
----	----	--	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

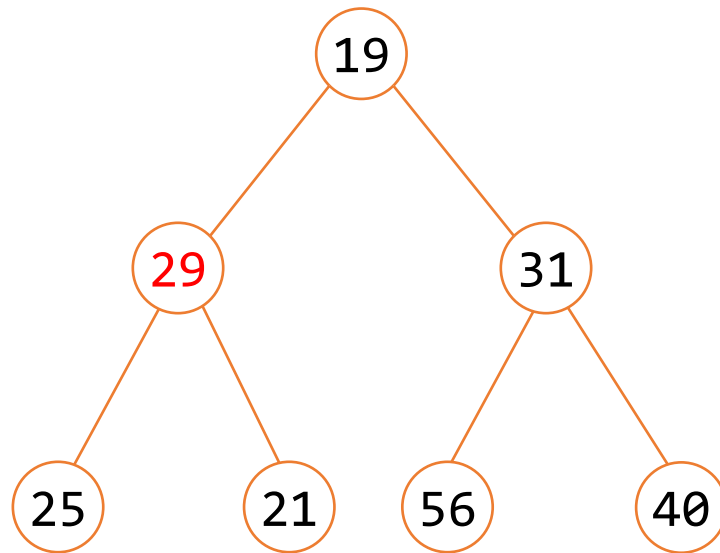
output

12	16			
----	----	--	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

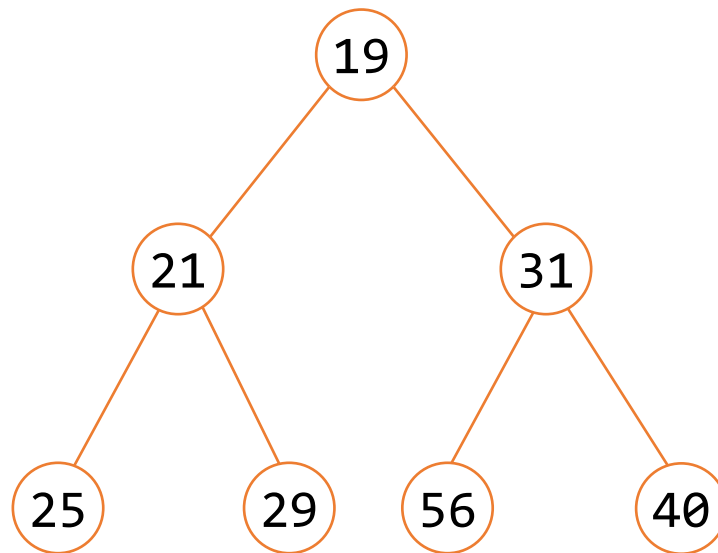
output

12	16			
----	----	--	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

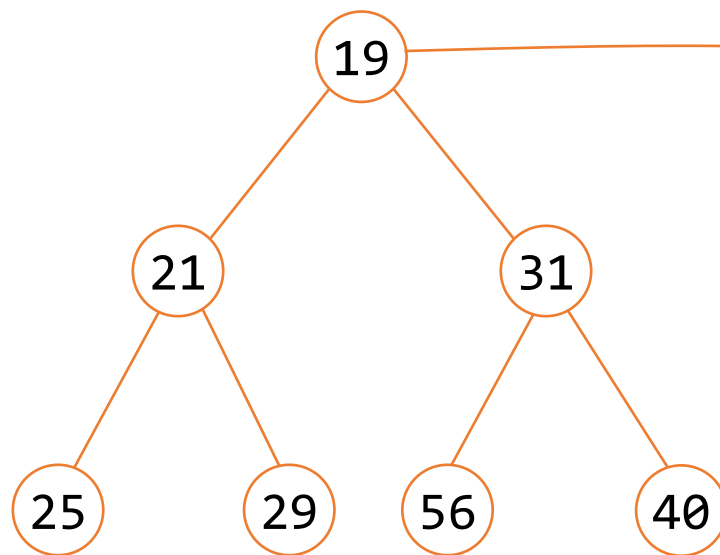
output

12	16			
----	----	--	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

output

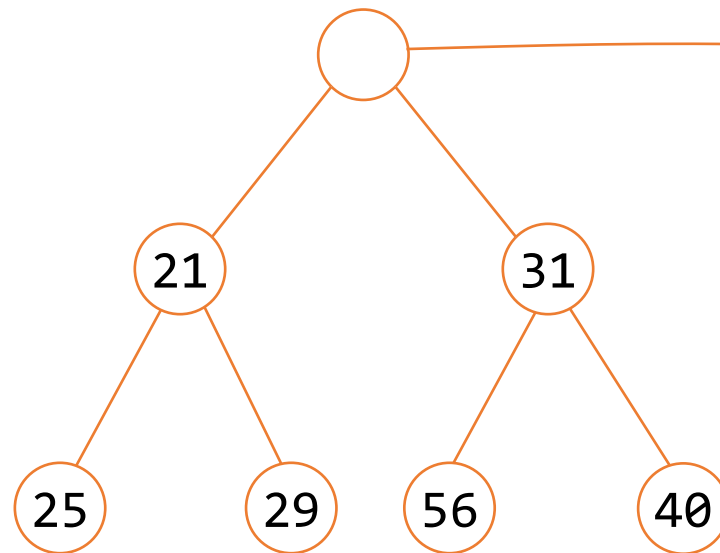
12	16			
----	----	--	--	--



# Пример работы

input

		14	35	23
--	--	----	----	----



**min-куча**

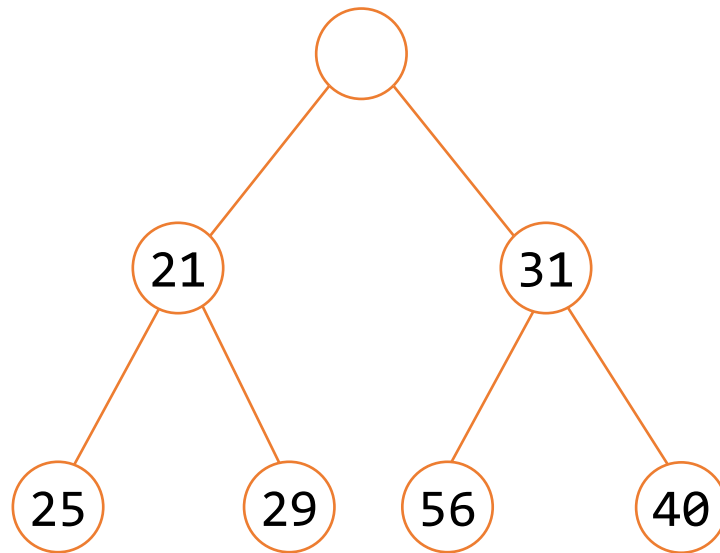
output

12	16	19		
----	----	----	--	--

# Пример работы

input

		14	35	23
--	--	----	----	----

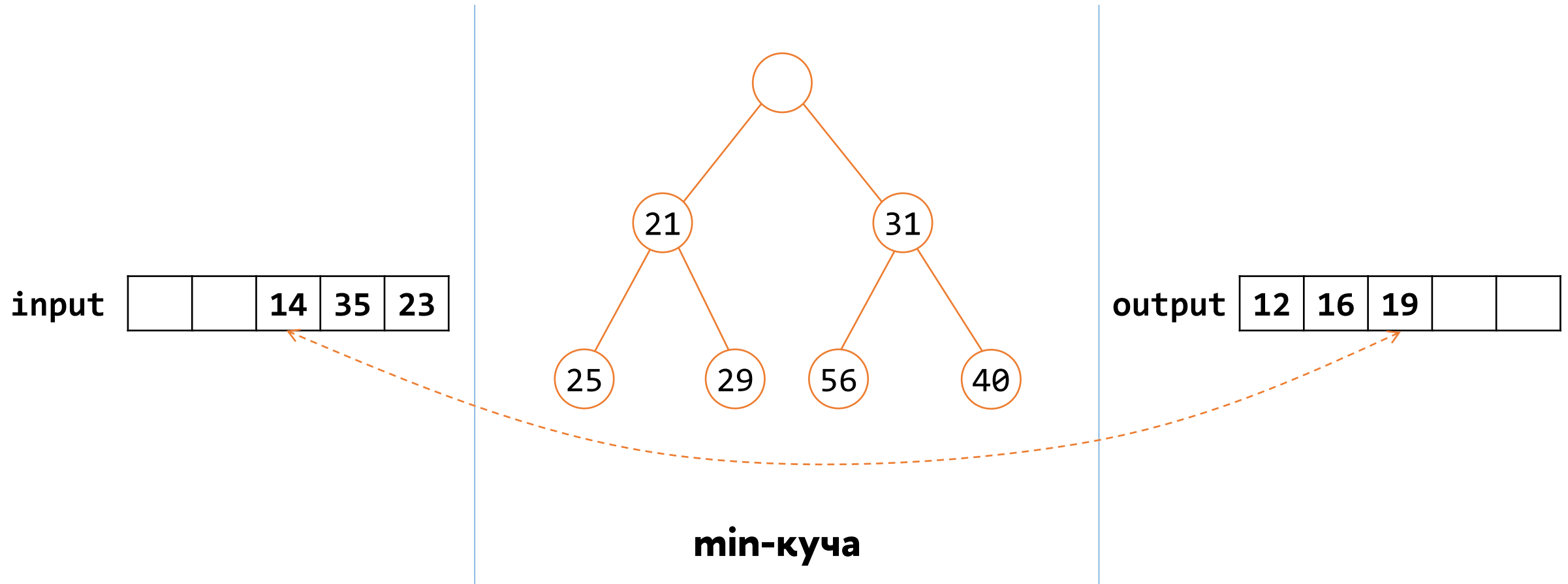


**min-куча**

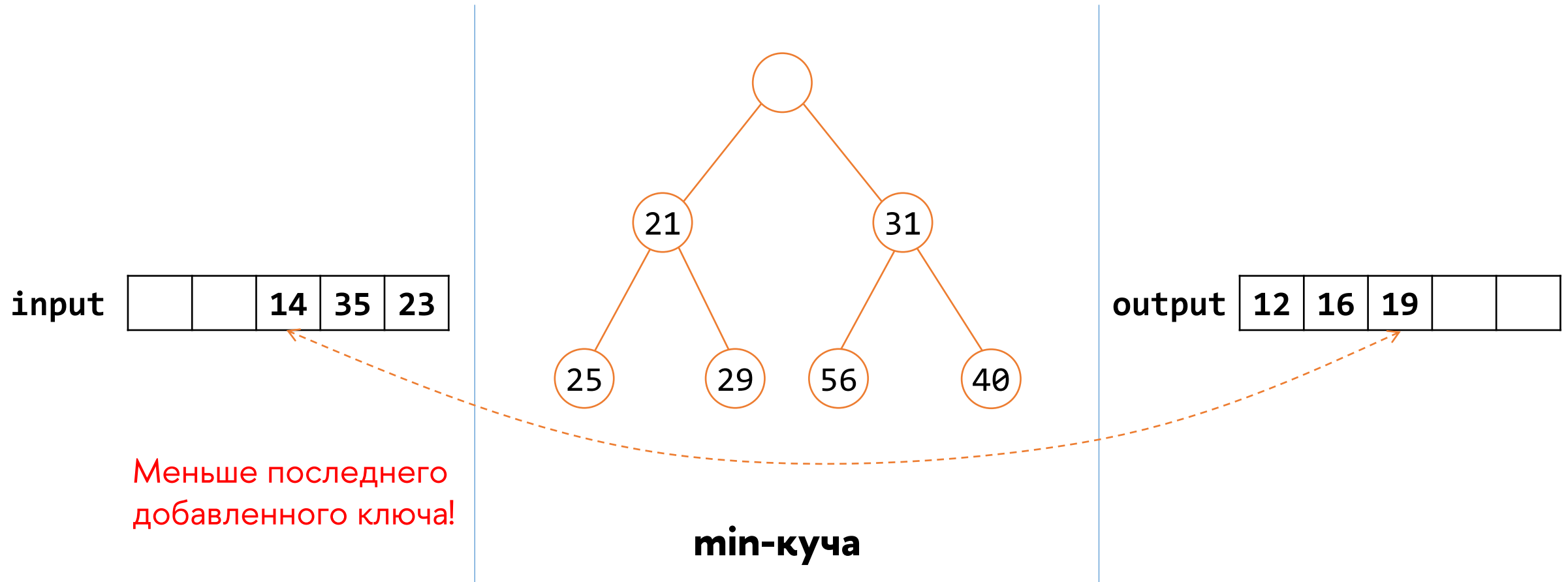
output

12	16	19		
----	----	----	--	--

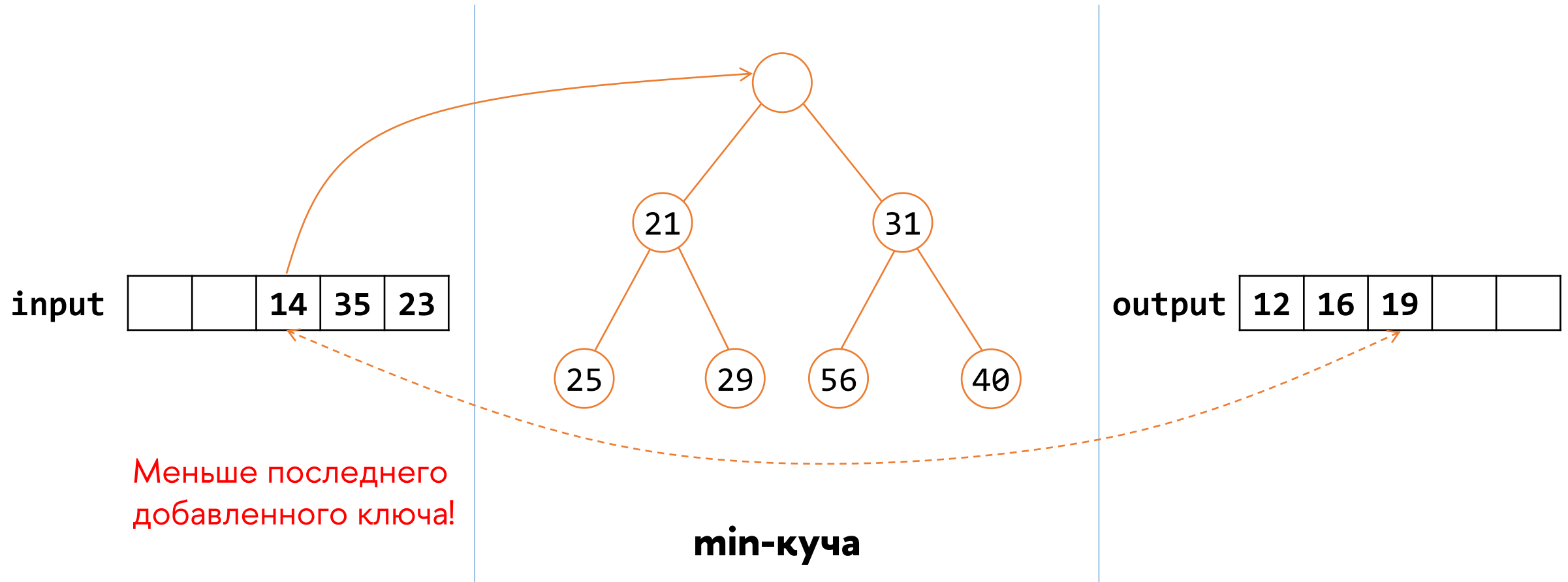
# Пример работы



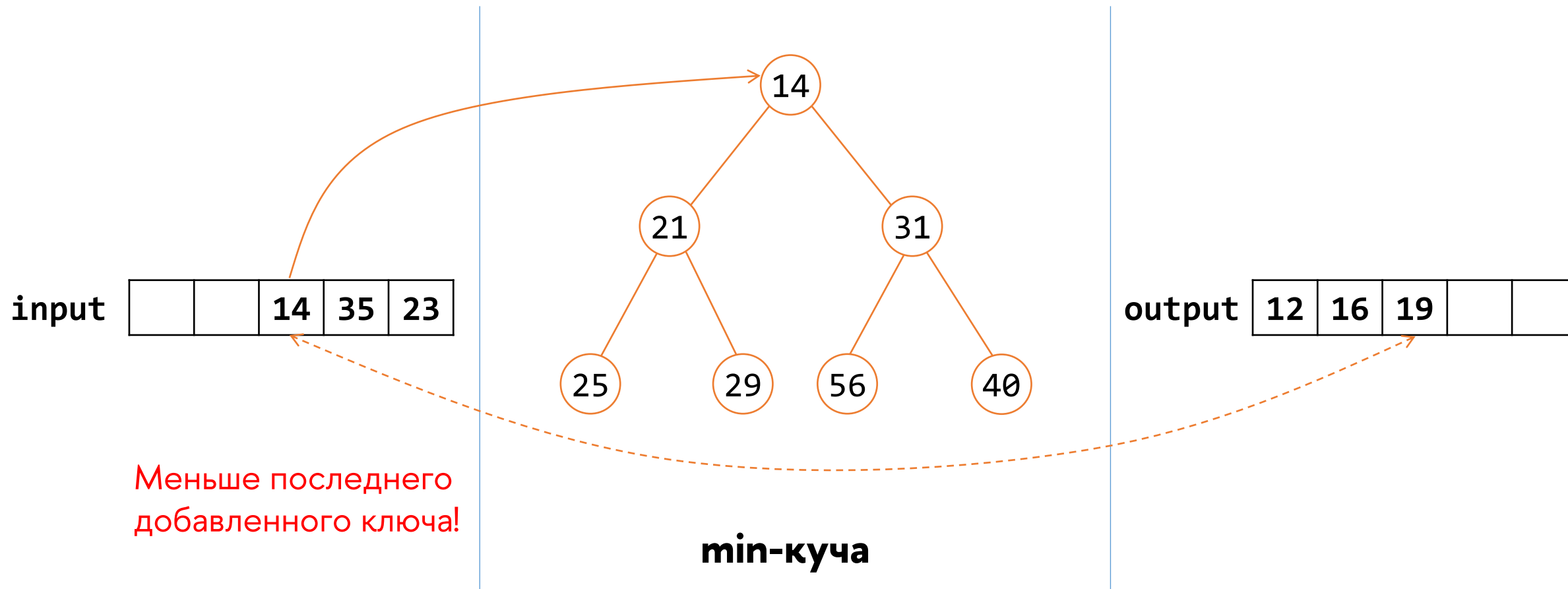
# Пример работы



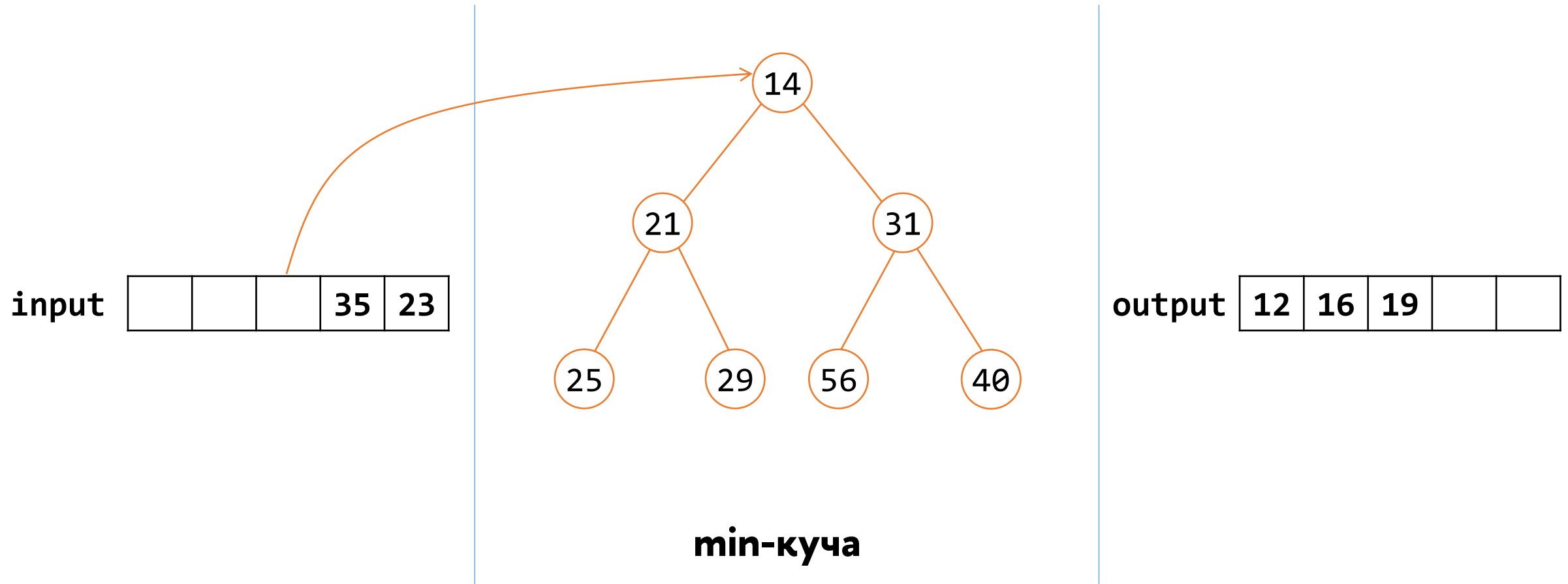
# Пример работы



# Пример работы



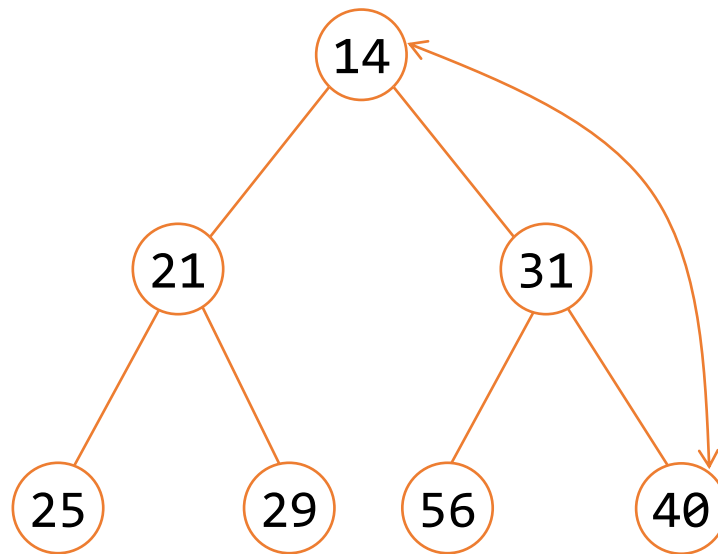
# Пример работы



# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

output

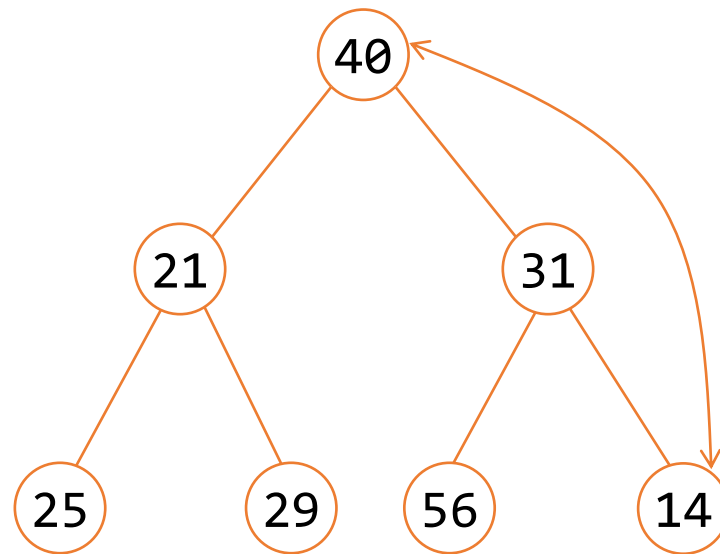
12	16	19		
----	----	----	--	--



# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

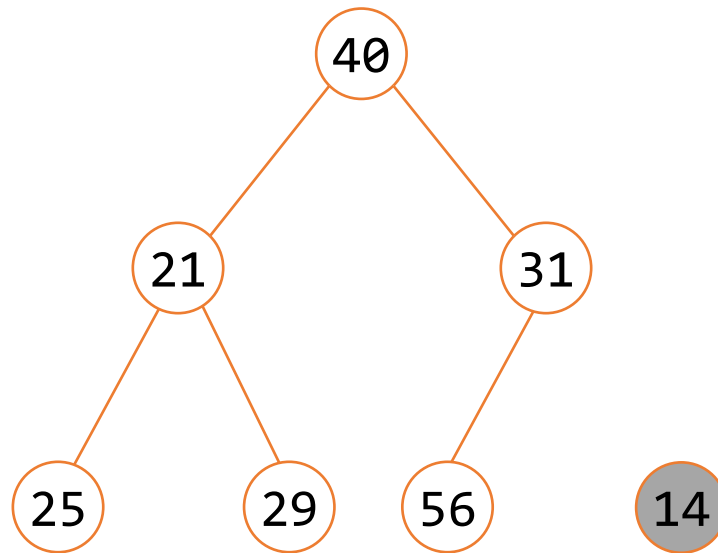
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

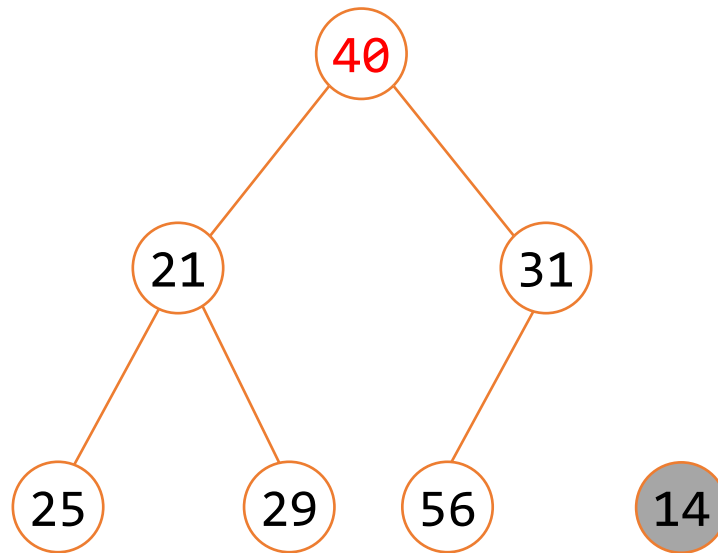
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

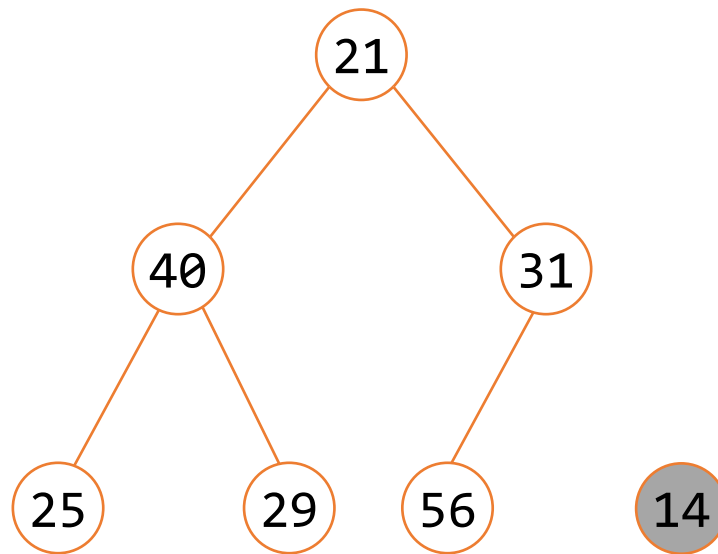
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

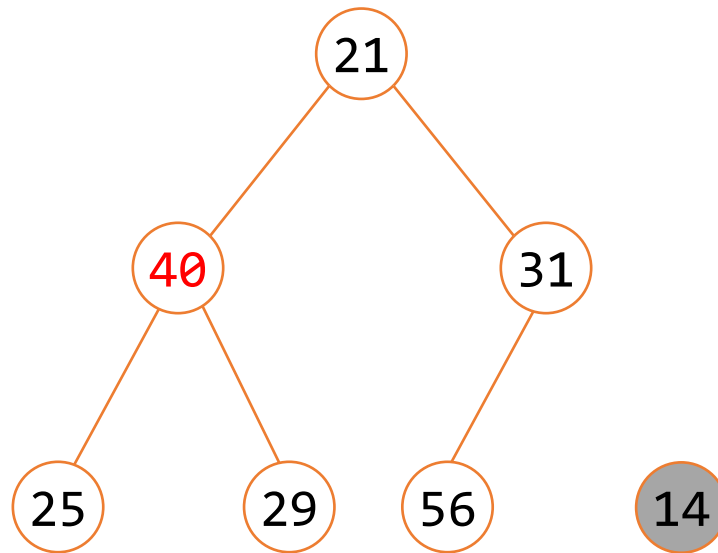
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



min-куча

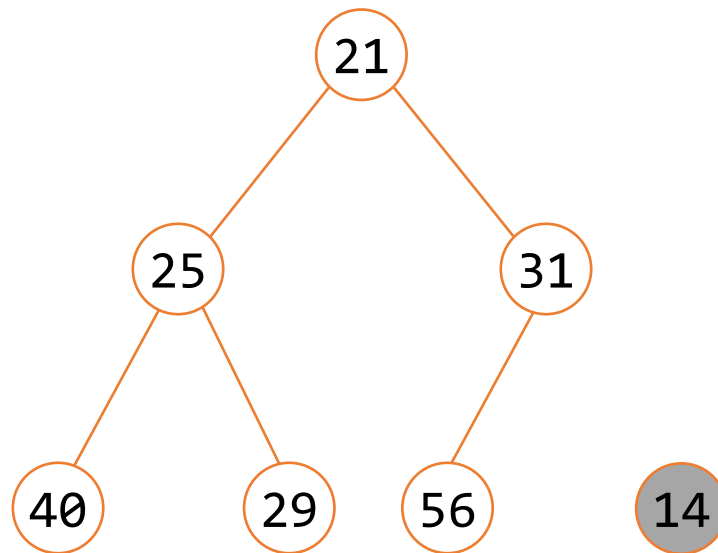
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

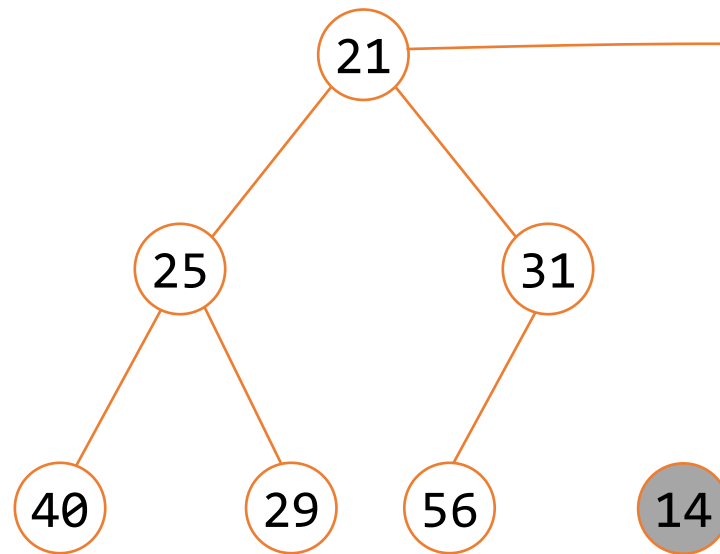
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

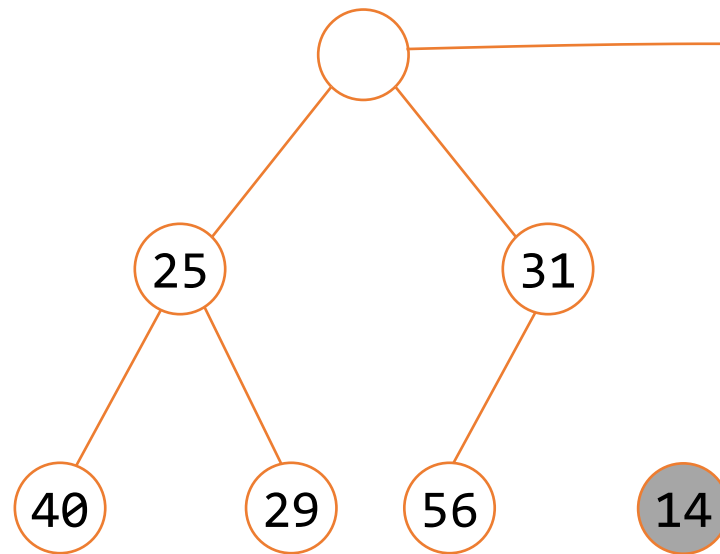
output

12	16	19		
----	----	----	--	--

# Пример работы

input

			35	23
--	--	--	----	----



**min-куча**

output

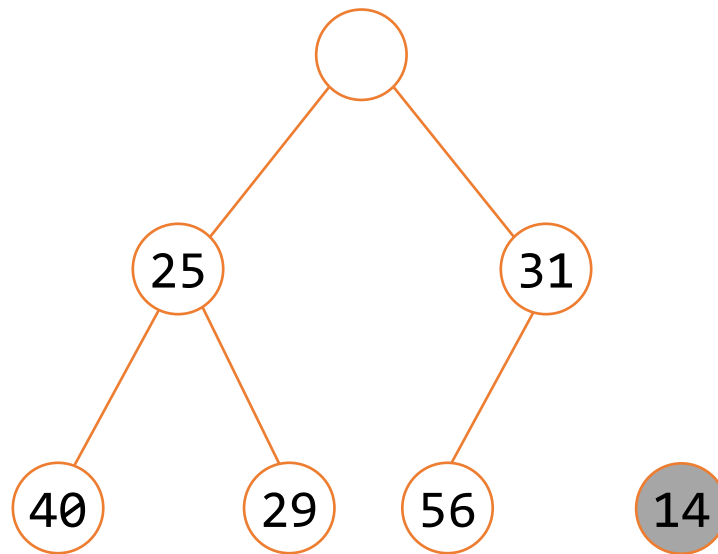
12	16	19	21	
----	----	----	----	--



# Пример работы

input

			35	23
--	--	--	----	----

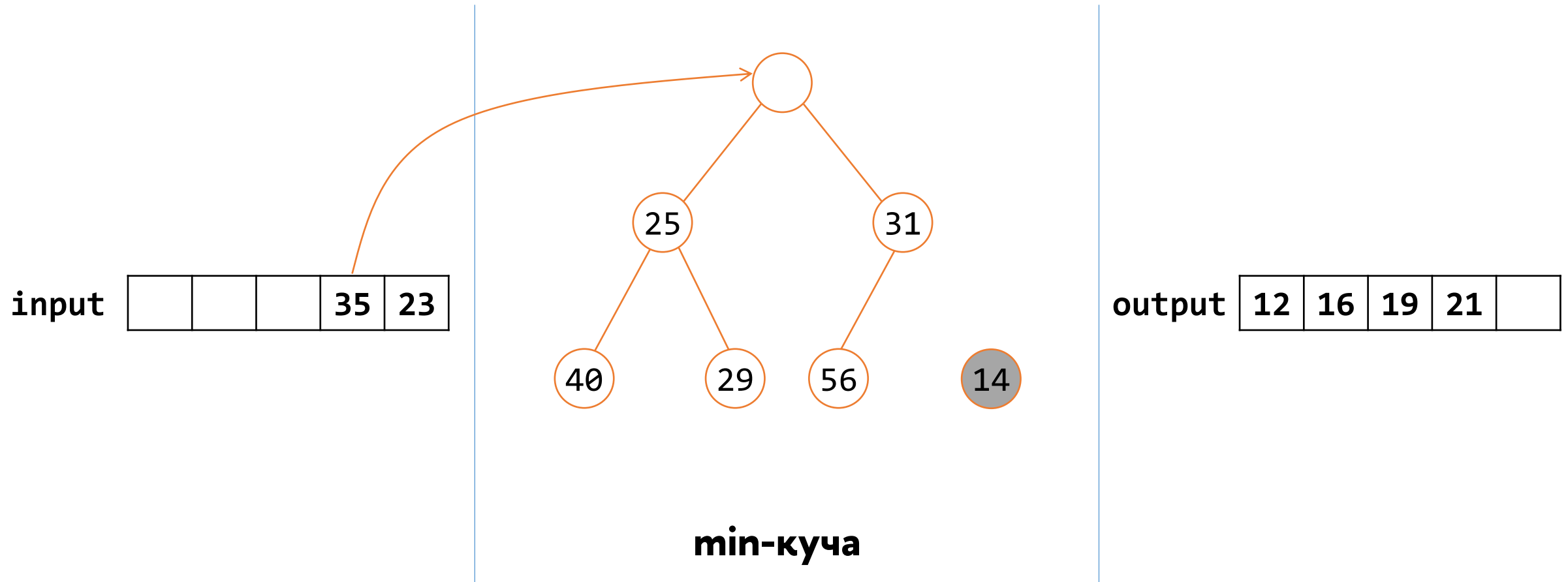


**min-куча**

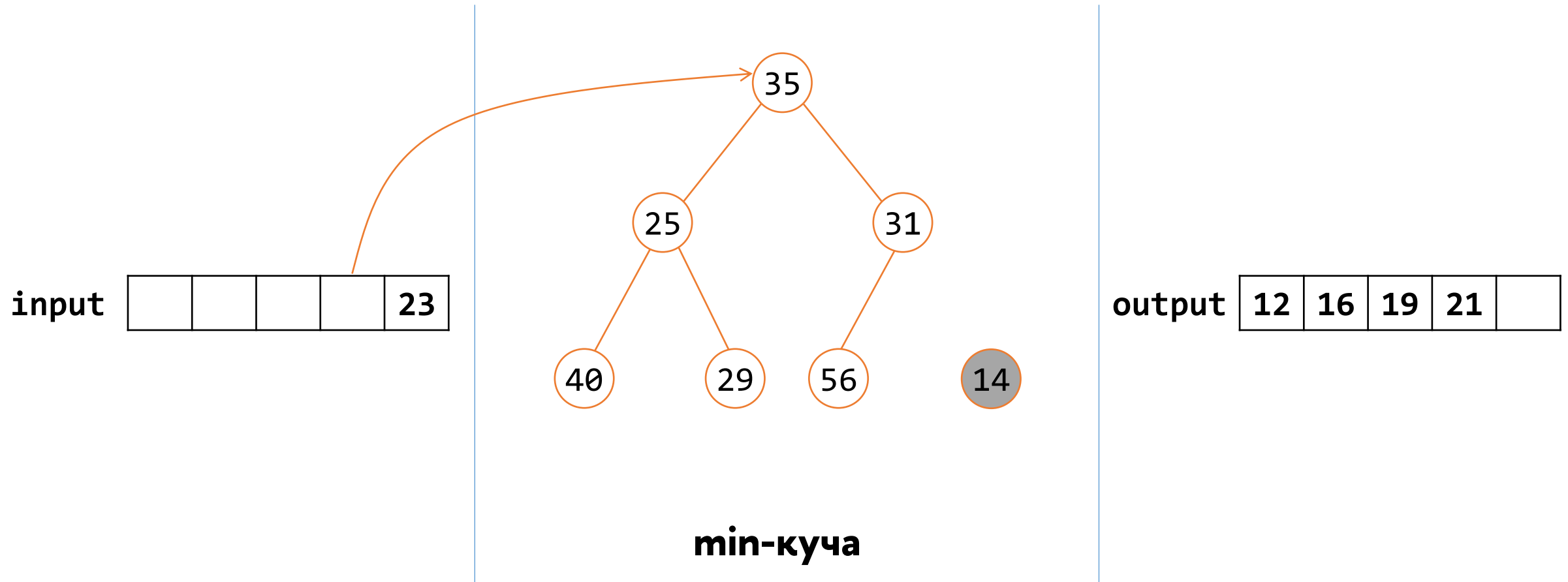
output

12	16	19	21	
----	----	----	----	--

# Пример работы

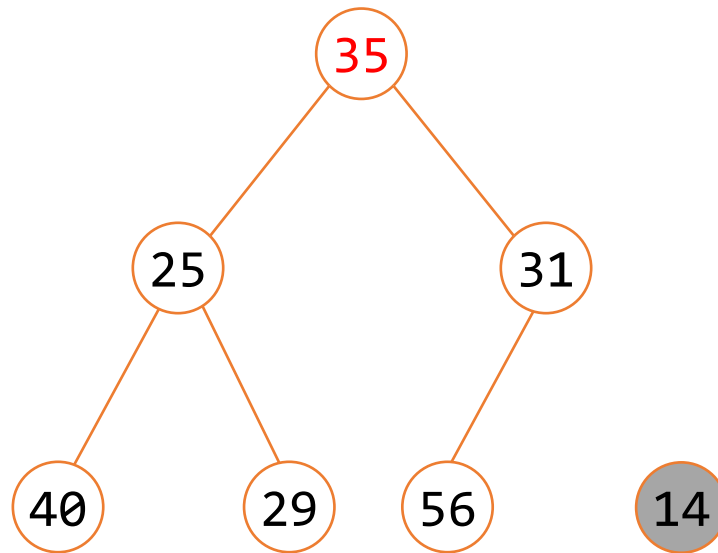


# Пример работы



# Пример работы

input



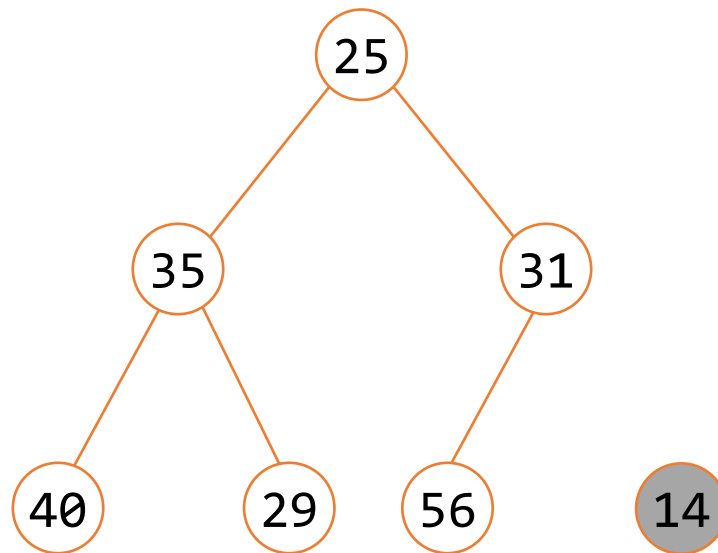
**min-куча**

output



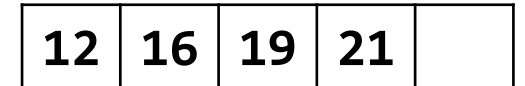
# Пример работы

input



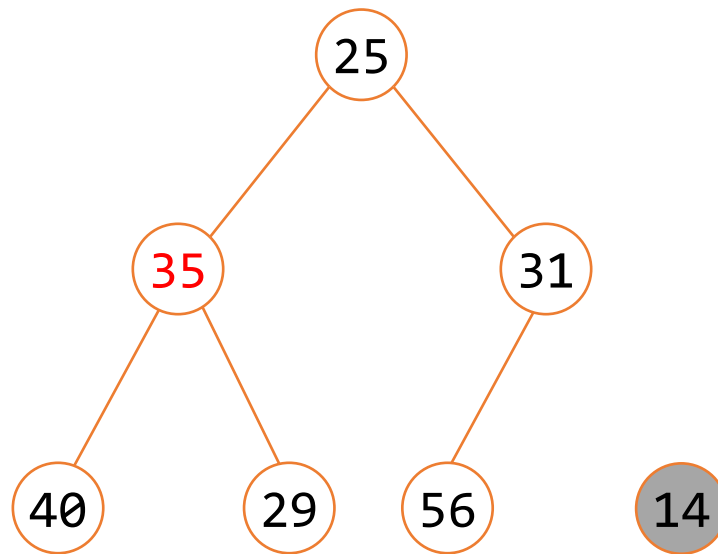
**min-куча**

output



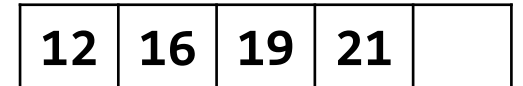
# Пример работы

input



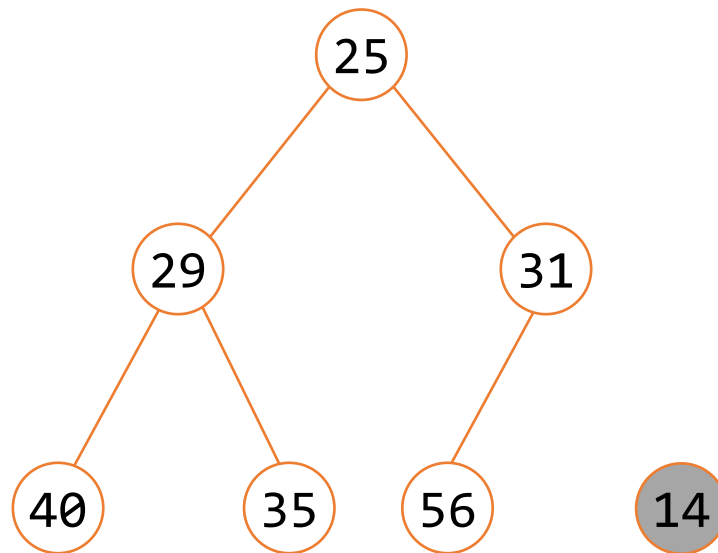
**min-куча**

output



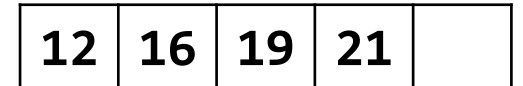
# Пример работы

input



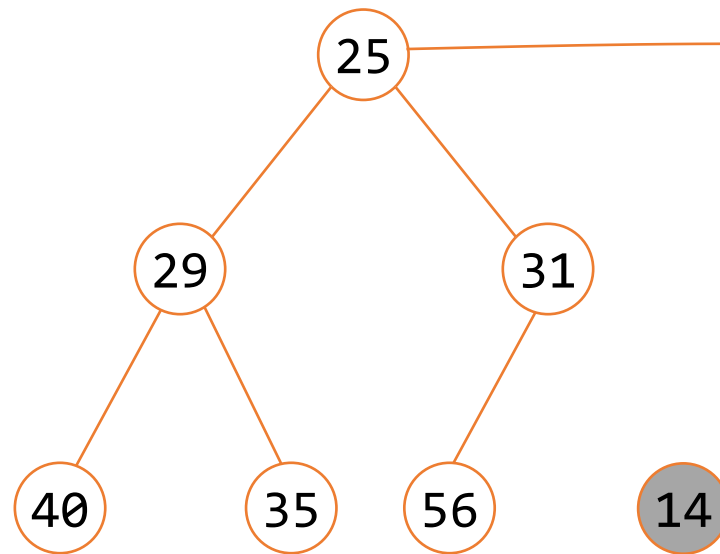
**min-куча**

output



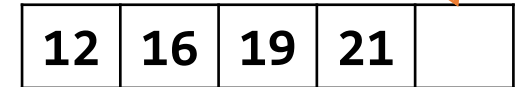
# Пример работы

input



**min-куча**

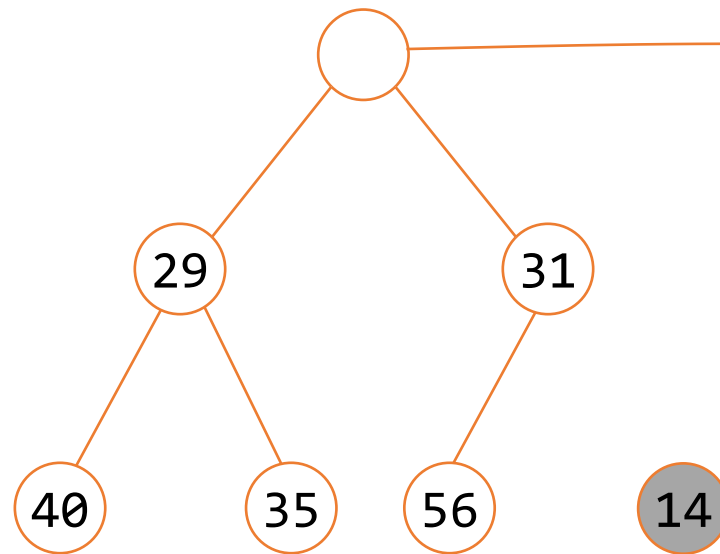
output





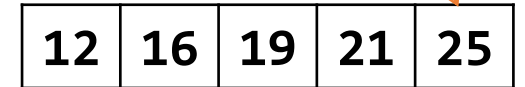
# Пример работы

input



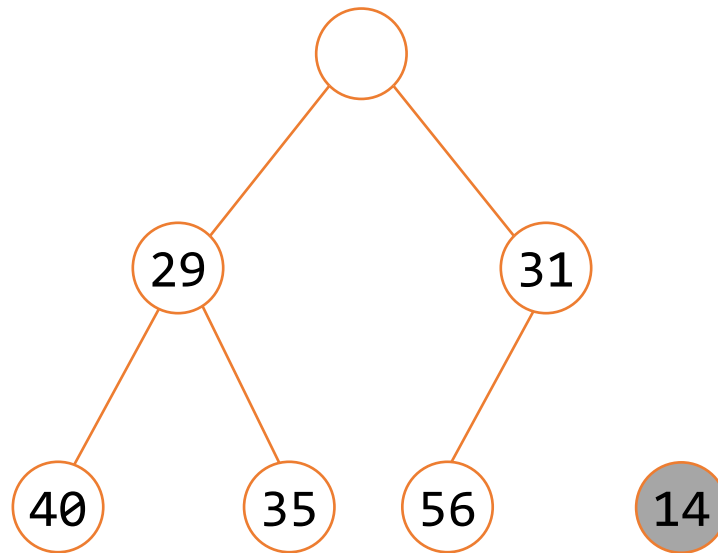
**min-куча**

output



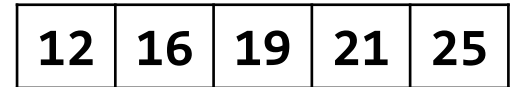
# Пример работы

input

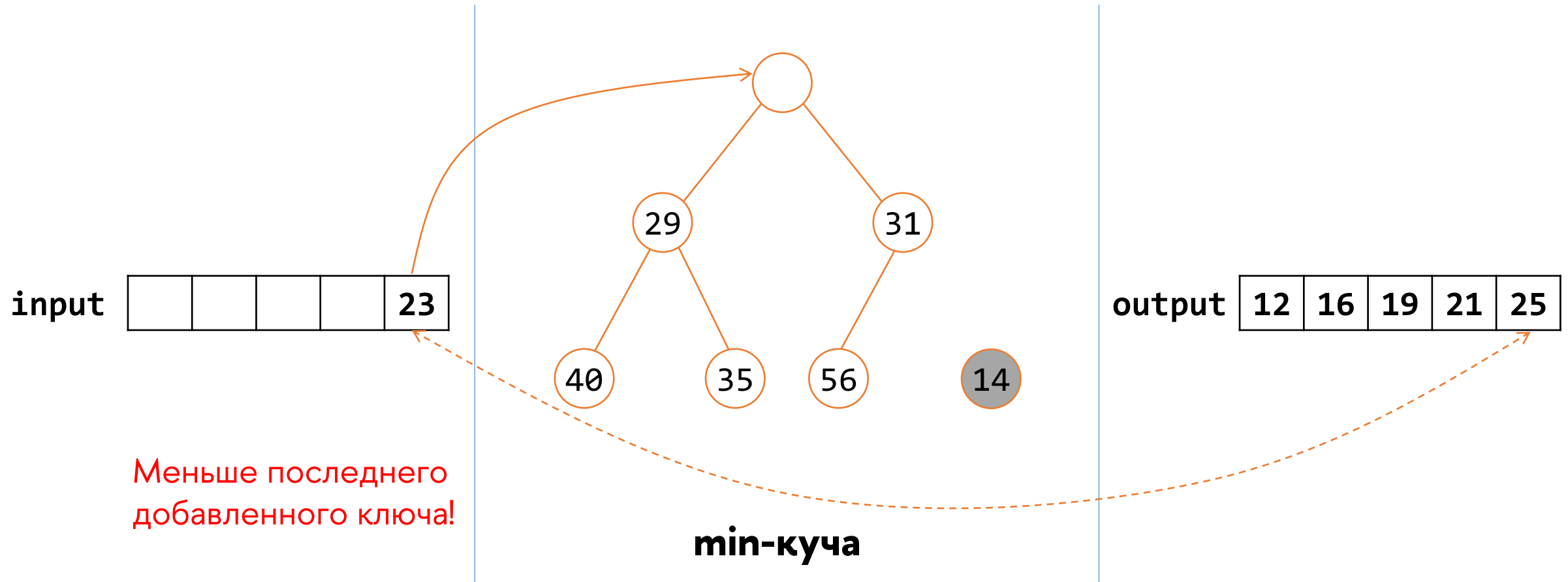


**min-куча**

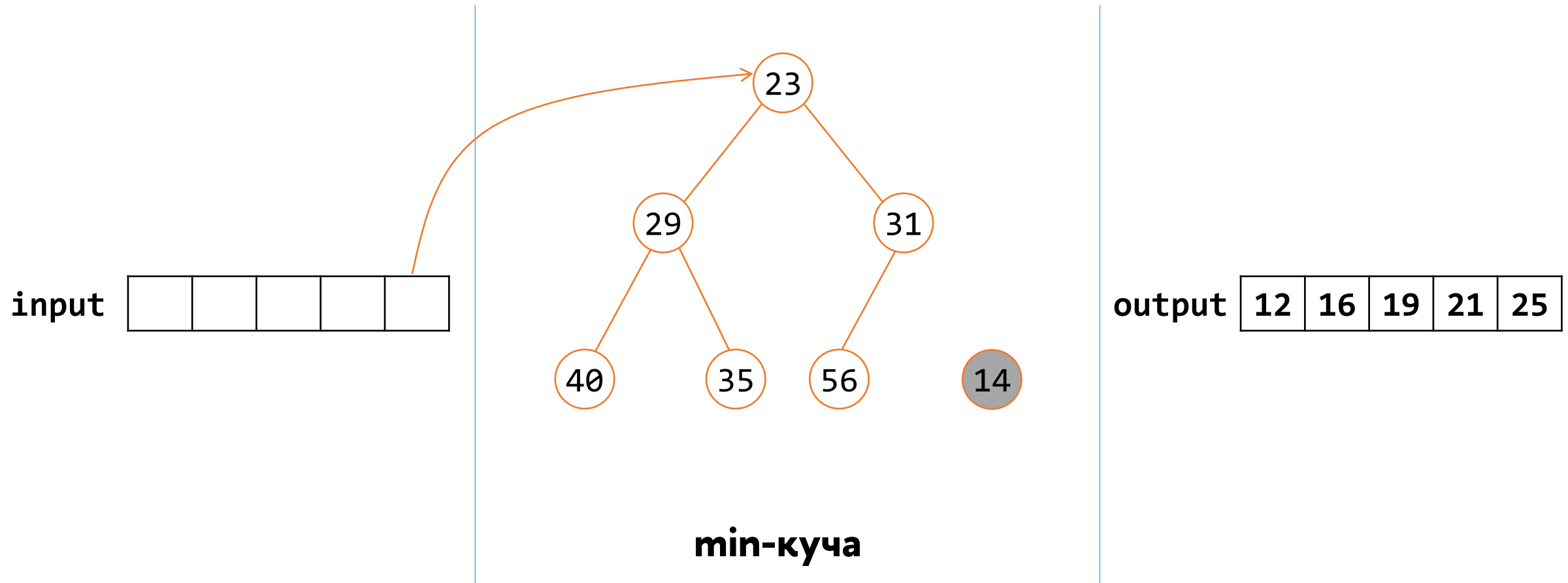
output



# Пример работы

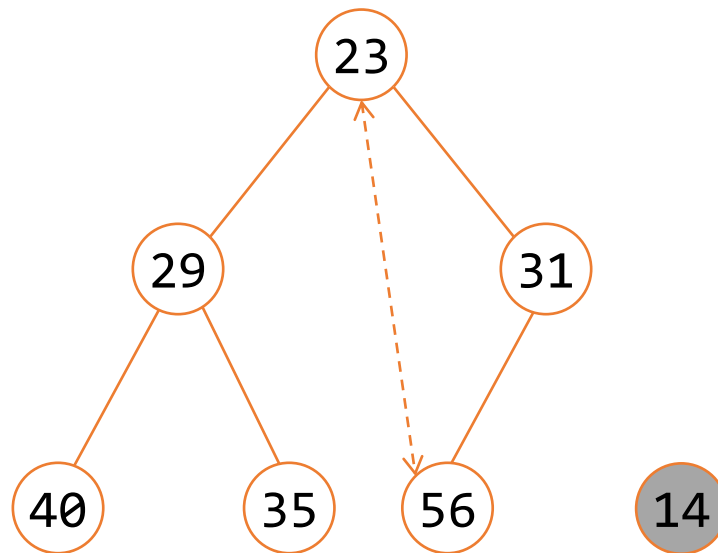


# Пример работы



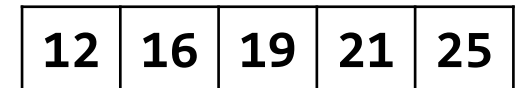
# Пример работы

input



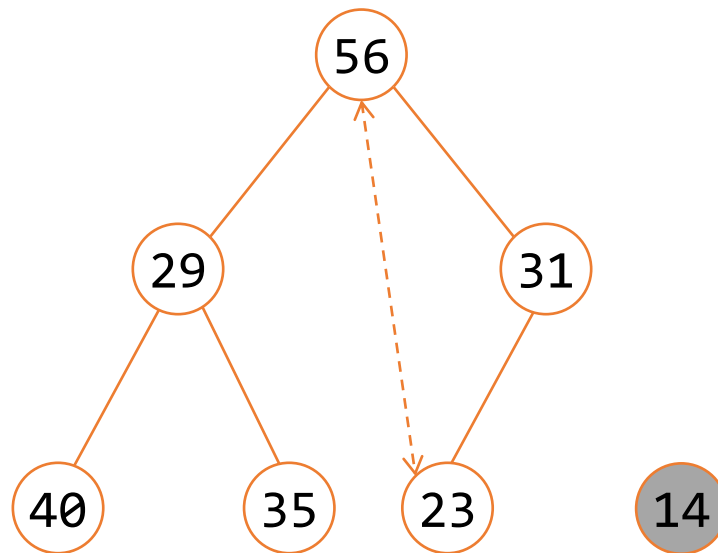
**min-куча**

output



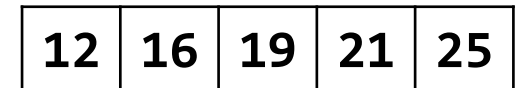
# Пример работы

input



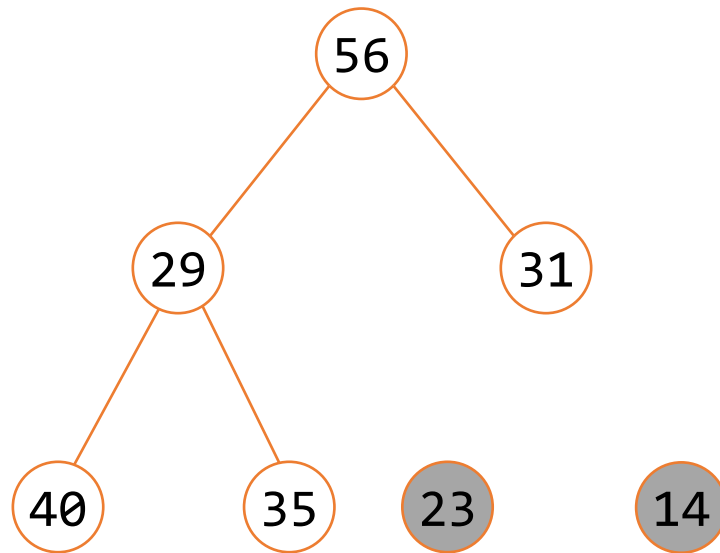
**min-куча**

output



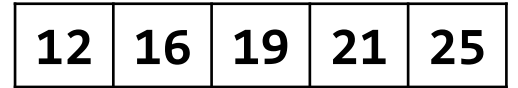
# Пример работы

input



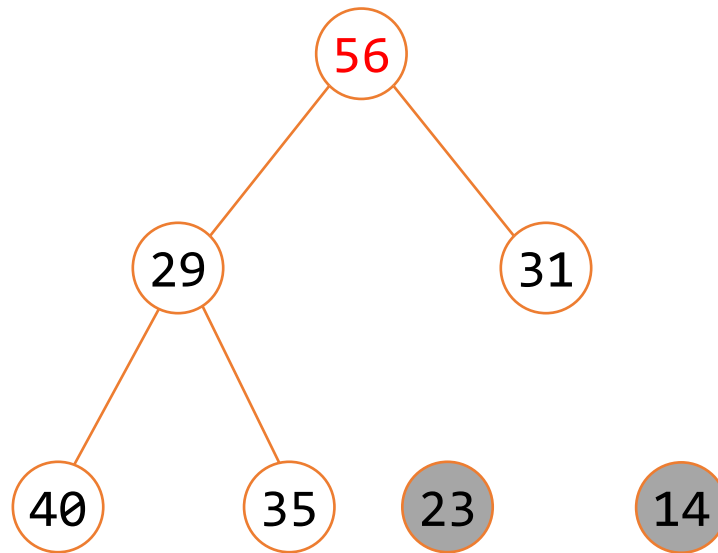
**min-куча**

output



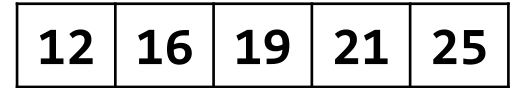
# Пример работы

input



**min-куча**

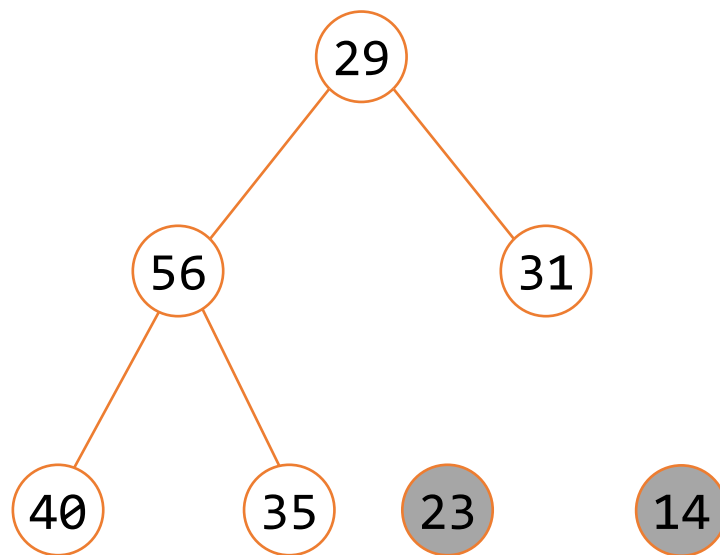
output





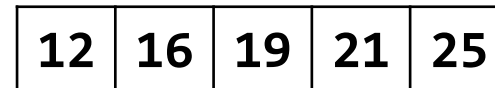
# Пример работы

input



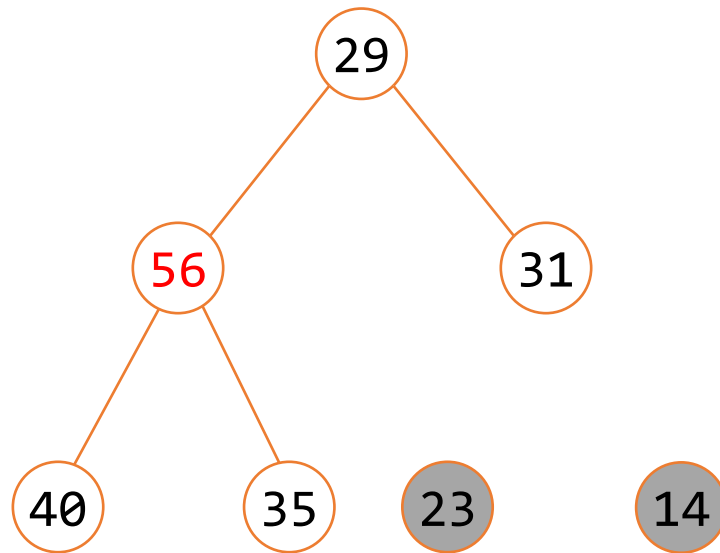
**min-куча**

output



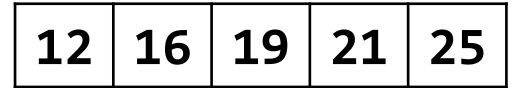
# Пример работы

input



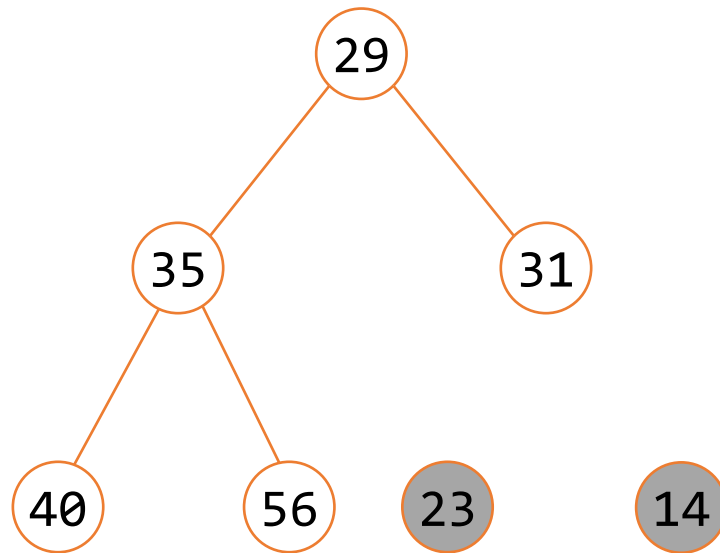
min-куча

output



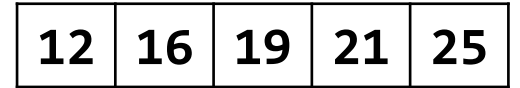
# Пример работы

input



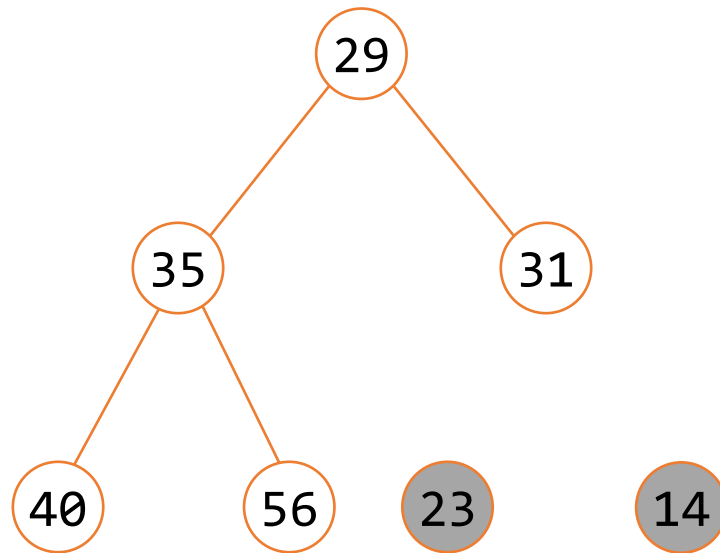
**min-куча**

output



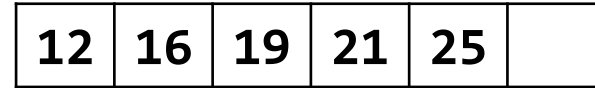
# Пример работы

input



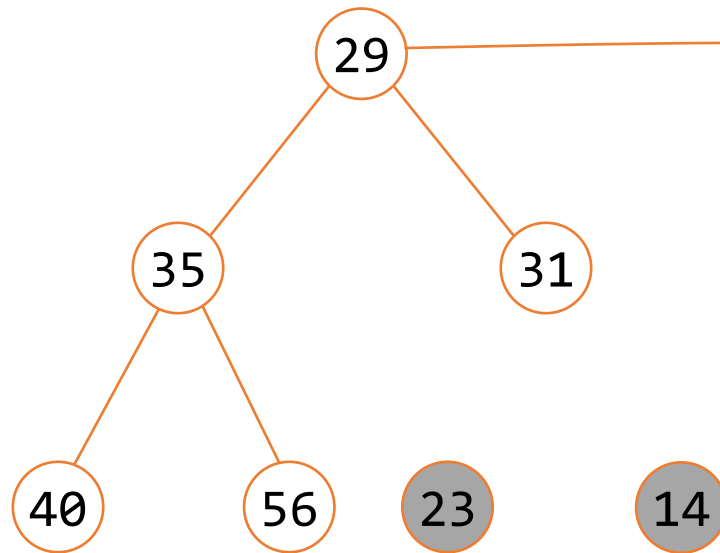
**min-куча**

output



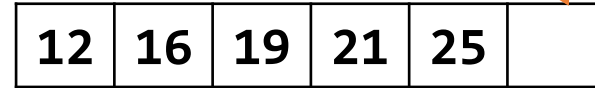
# Пример работы

input



**min-куча**

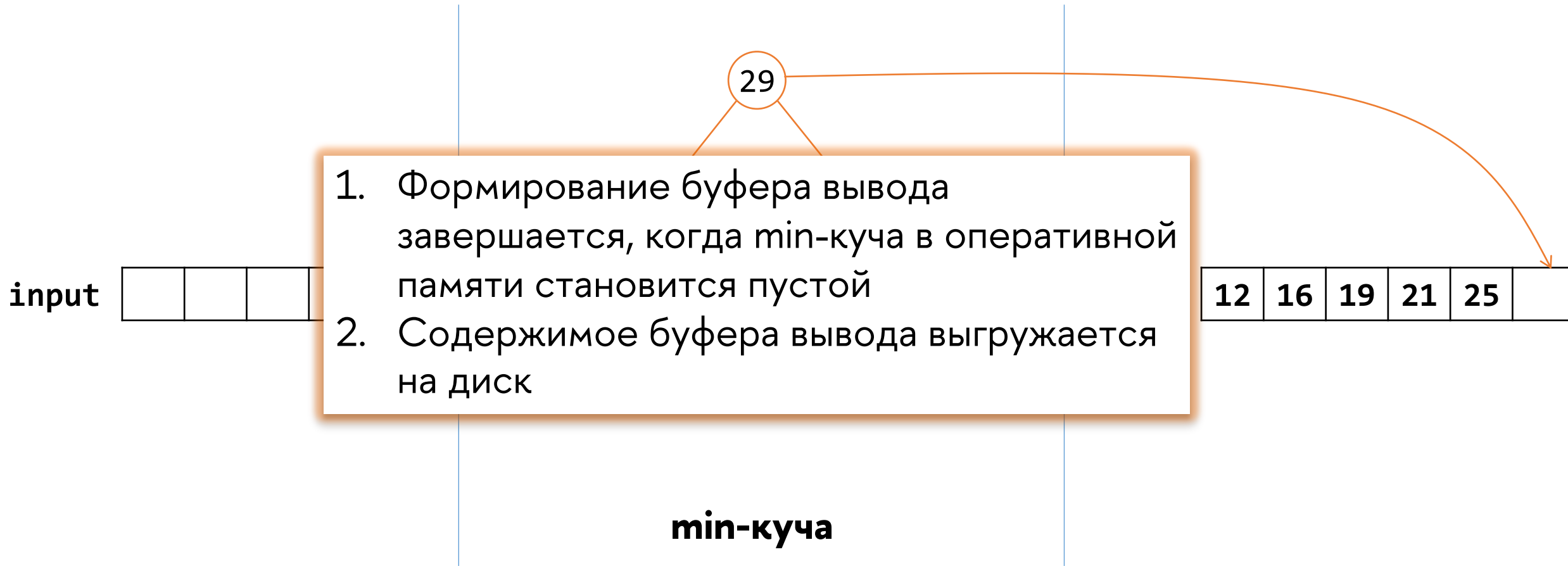
output



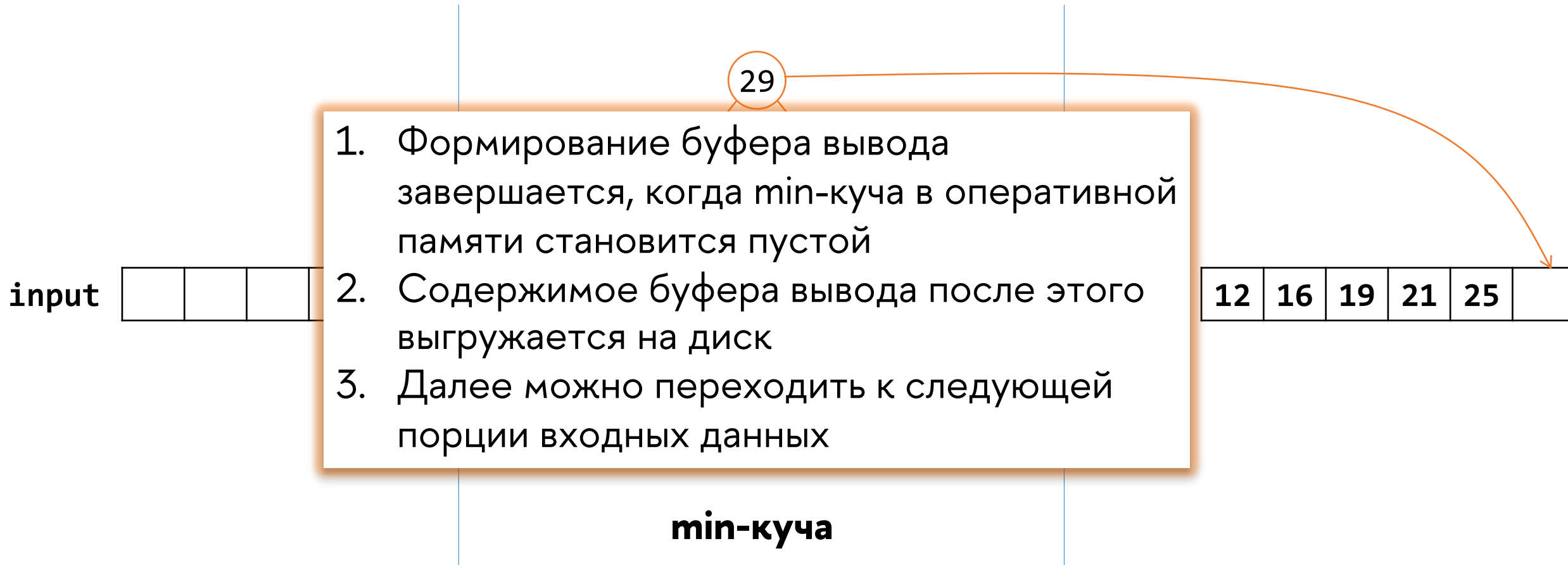
# Пример работы



# Пример работы



# Пример работы





После того, как мы подготовили  
отсортированные фрагменты входных  
данных, выполняем  
*многопроходное слияние*

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

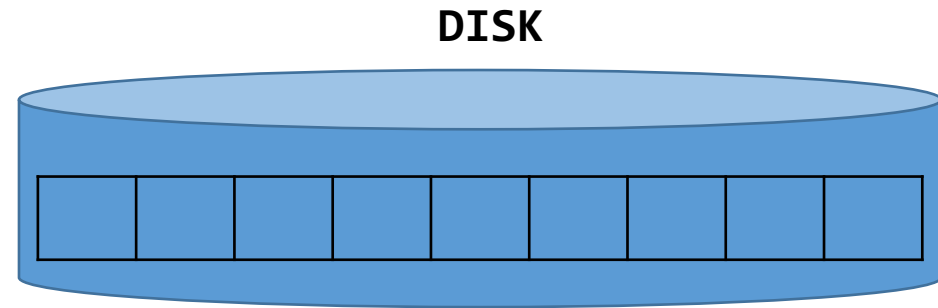
39	64	79
----	----	----

input2 

20	88	99
----	----	----

input3 

17	29	94
----	----	----



OUTPUT 

--	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

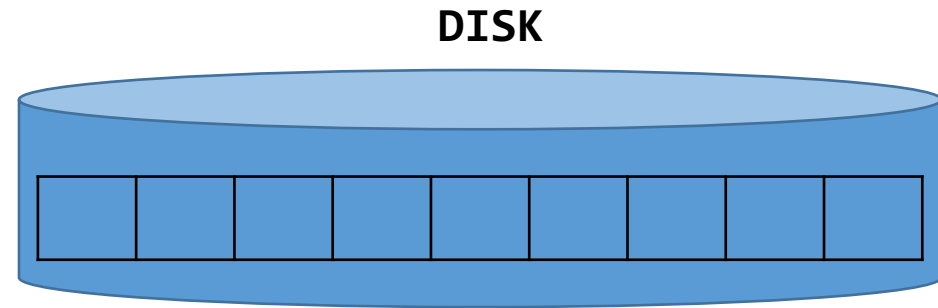
39	64	79
----	----	----

input2 

20	88	99
----	----	----

input3 

17	29	94
----	----	----



OUTPUT 

--	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

39	64	79
----	----	----

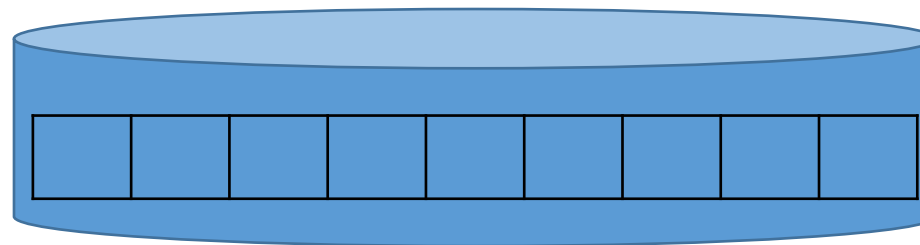
input2 

20	88	99
----	----	----

input3 

17	29	94
----	----	----

DISK



OUTPUT



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

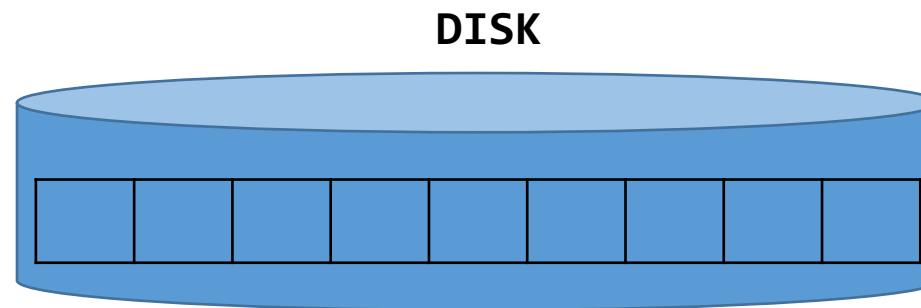
39	64	79
----	----	----

input2 

20	88	99
----	----	----

input3 

	29	94
--	----	----



OUTPUT 

17		
----	--	--



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

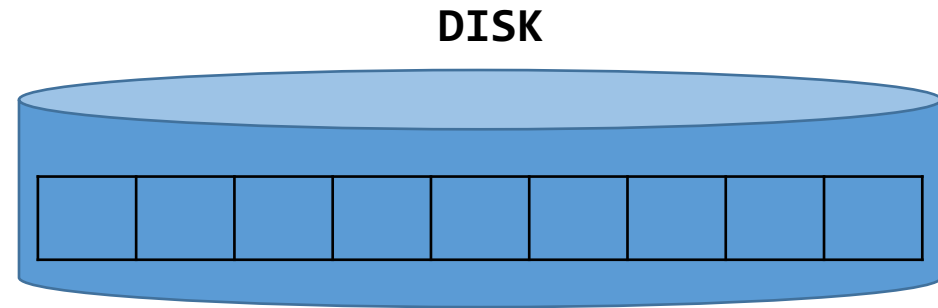
39	64	79
----	----	----

input2 

20	88	99
----	----	----

input3 

	29	94
--	----	----



OUTPUT 

17		
----	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

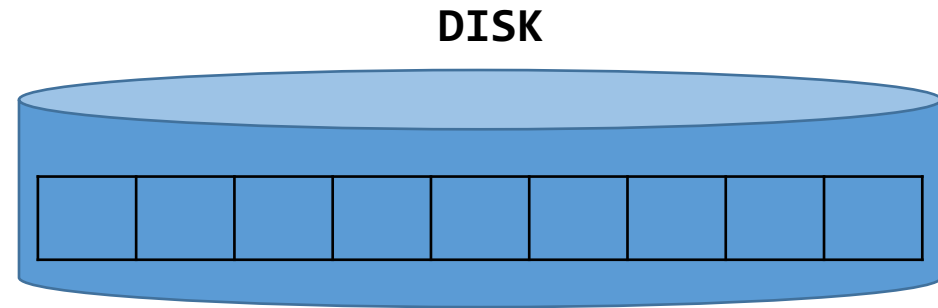
39	64	79
----	----	----

input2 

20	88	99
----	----	----

input3 

	29	94
--	----	----



OUTPUT 

17		
----	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

39	64	79
----	----	----

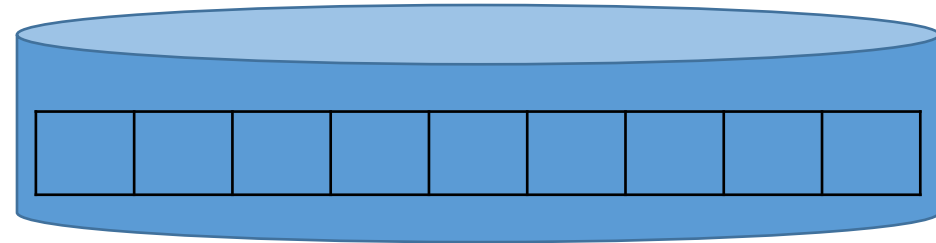
input2 

20	88	99
----	----	----

input3 

	29	94
--	----	----

DISK



OUTPUT 

17		
----	--	--





# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

39	64	79
----	----	----

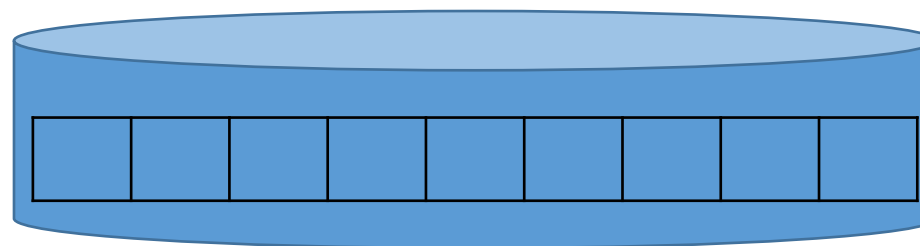
input2 

	88	99
--	----	----

input3 

	29	94
--	----	----

DISK



OUTPUT 

17	20	
----	----	--



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

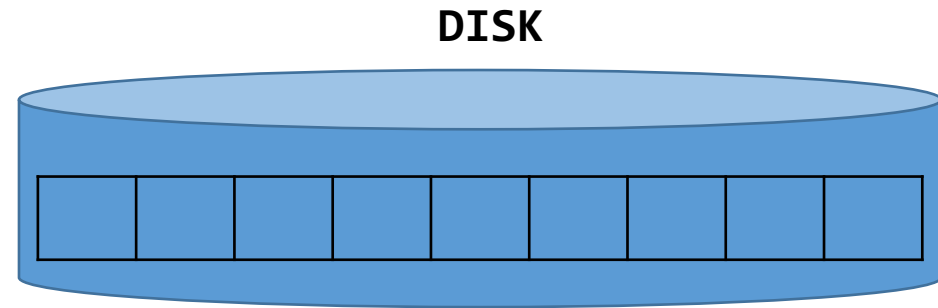
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

	29	94
--	----	----



OUTPUT 

17	20	
----	----	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

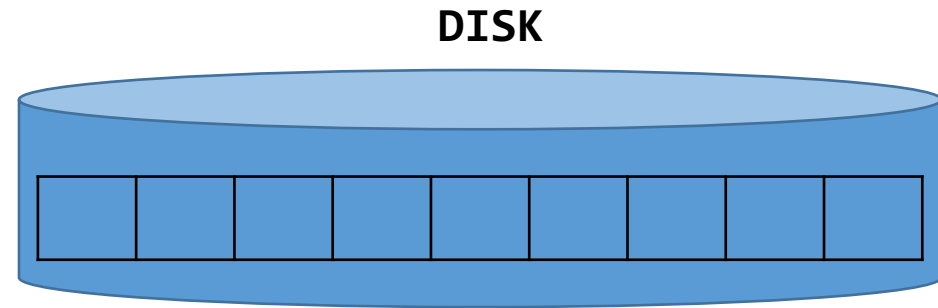
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

	29	94
--	----	----



OUTPUT 

17	20	
----	----	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

39	64	79
----	----	----

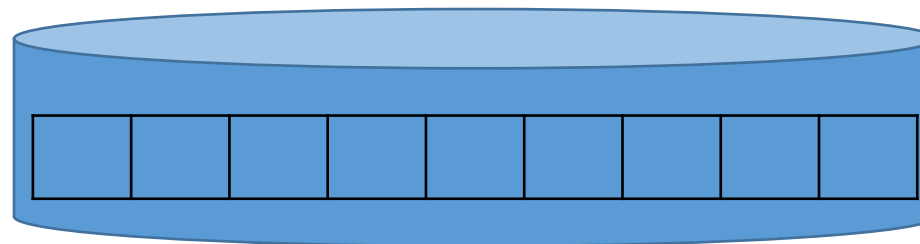
input2 

	88	99
--	----	----

input3 

	29	94
--	----	----

DISK



OUTPUT 

17	20	
----	----	--



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

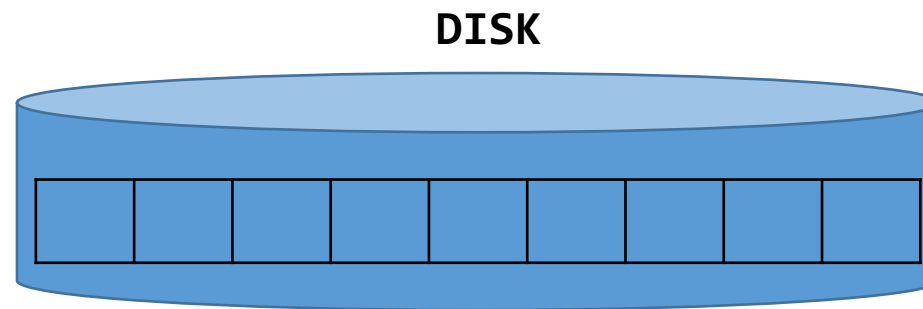
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

	29	94
--	----	----



OUTPUT 

17	20	29
----	----	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

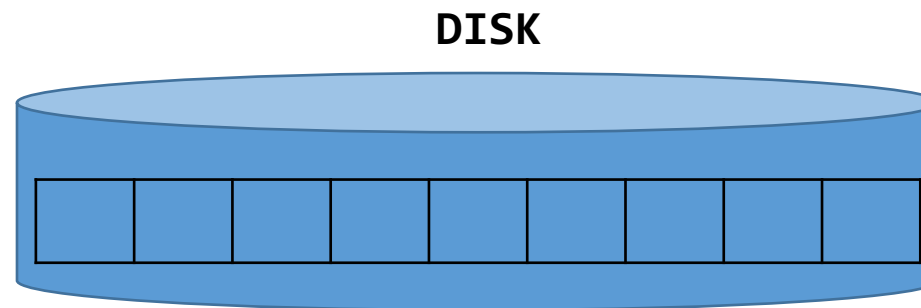
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

17	20	29
----	----	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

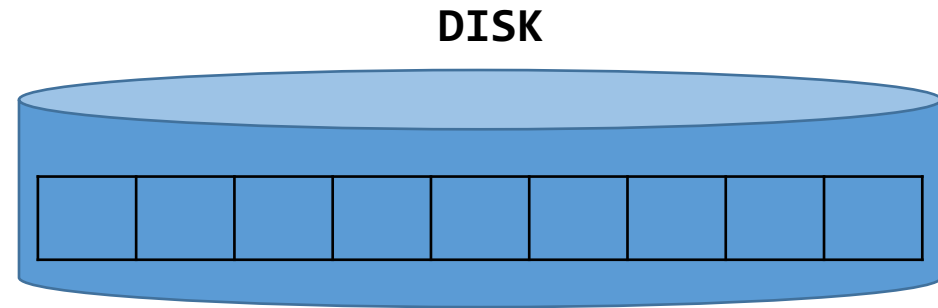
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

17	20	29
----	----	----

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

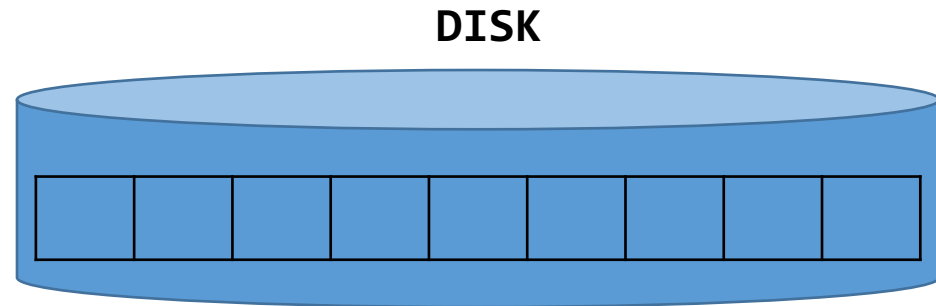
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

17	20	29
----	----	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

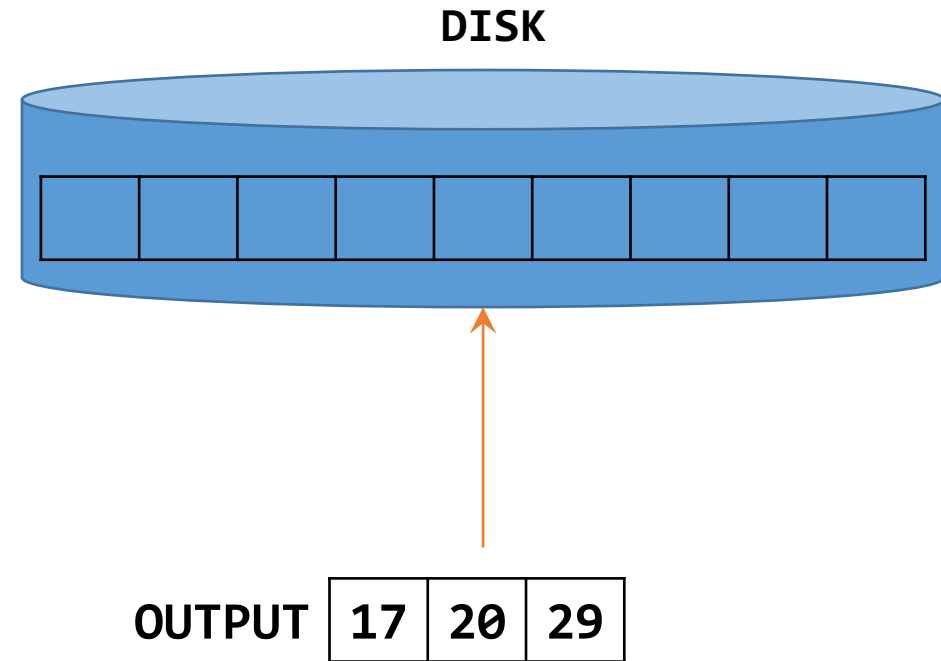
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

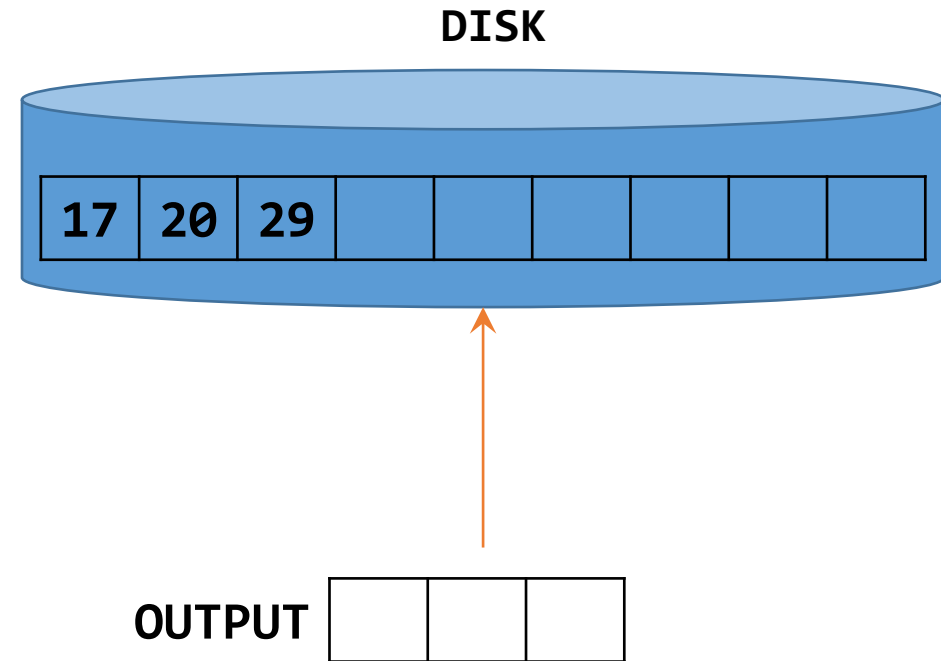
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

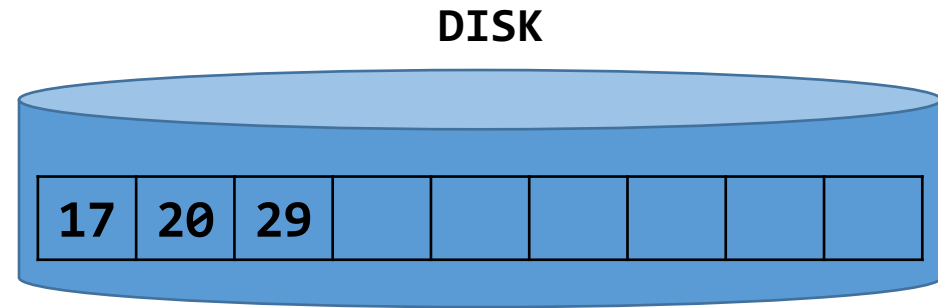
39	64	79
----	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

--	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

39	64	79
----	----	----

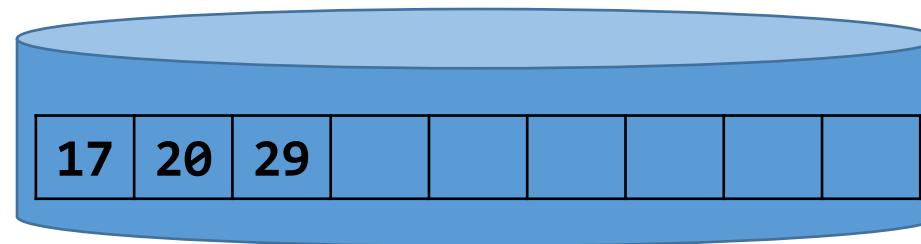
input2 

	88	99
--	----	----

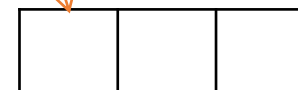
input3 

		94
--	--	----

DISK



OUTPUT



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

	64	79
--	----	----

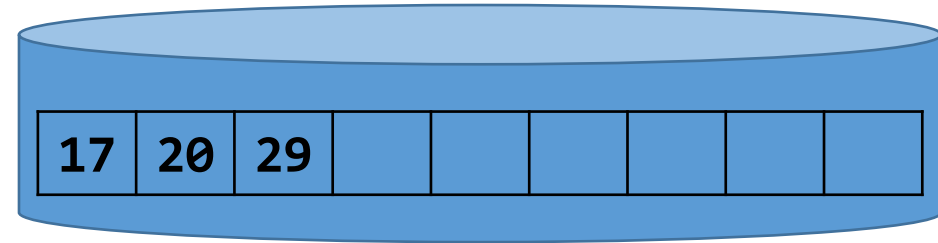
input2 

	88	99
--	----	----

input3 

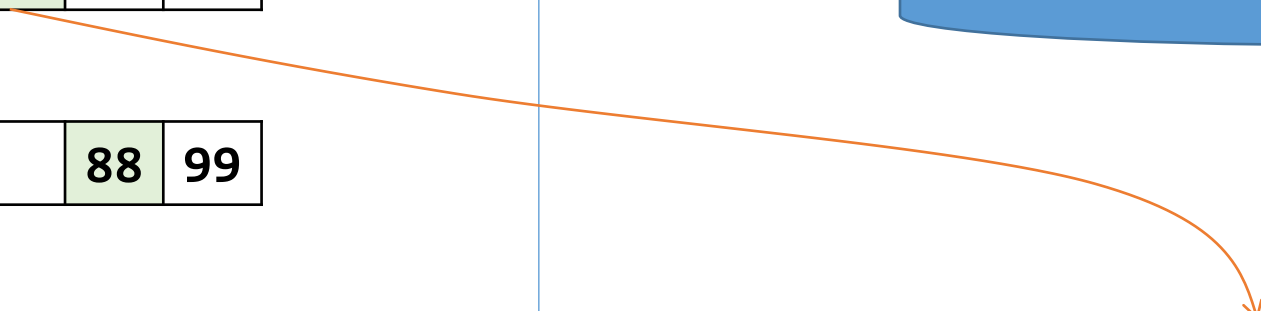
		94
--	--	----

DISK



OUTPUT 

39		
----	--	--



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

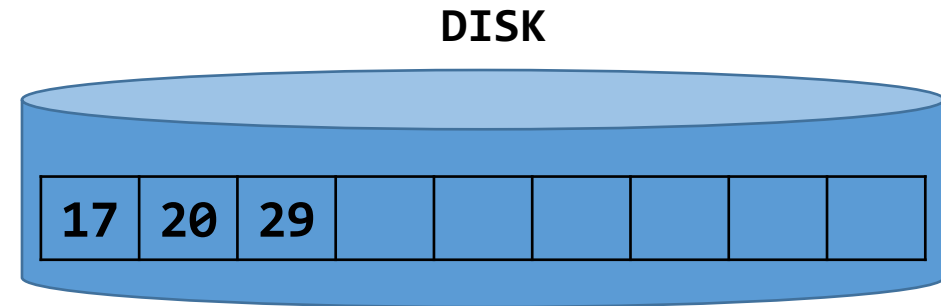
	64	79
--	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

39		
----	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

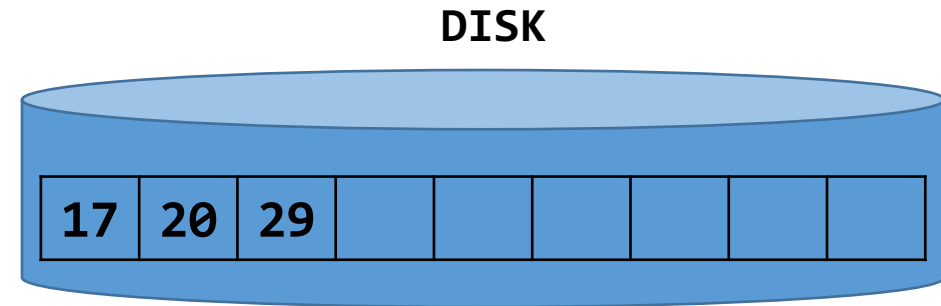
	64	79
--	----	----

input2 

	88	99
--	----	----

input3 

		94
--	--	----



OUTPUT 

39		
----	--	--

# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

--	--	--

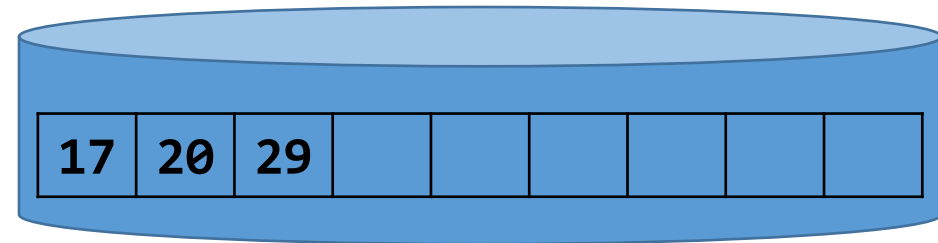
input2 

	88	99
--	----	----

input3 

		94
--	--	----

DISK



OUTPUT 

39	64	79
----	----	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

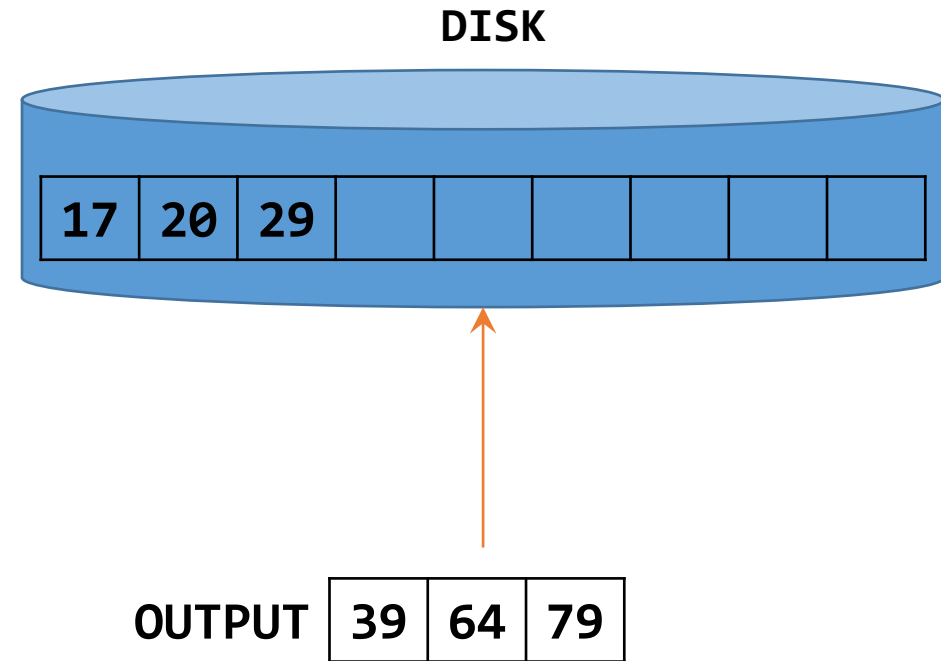
--	--	--

input2 

	88	99
--	----	----

input3 

		94
--	--	----



# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1 

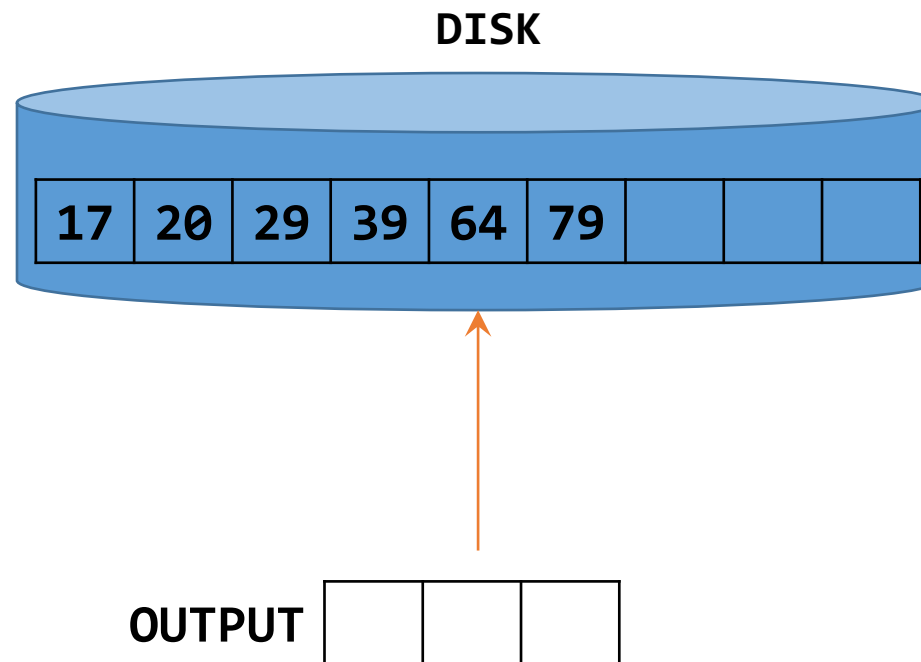
--	--	--

input2 

	88	99
--	----	----

input3 

		94
--	--	----



# Пример работы

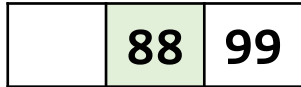
Фрагменты файлов, полученных  
на предыдущем шаге

input1



И так далее...

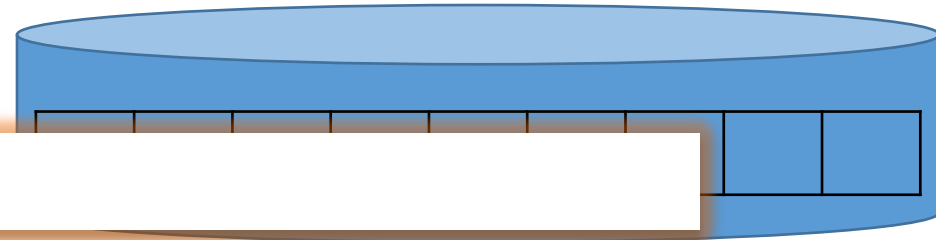
input2



input3



DISK



OUTPUT



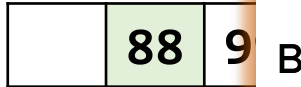
# Пример работы

Фрагменты файлов, полученных  
на предыдущем шаге

input1



input2



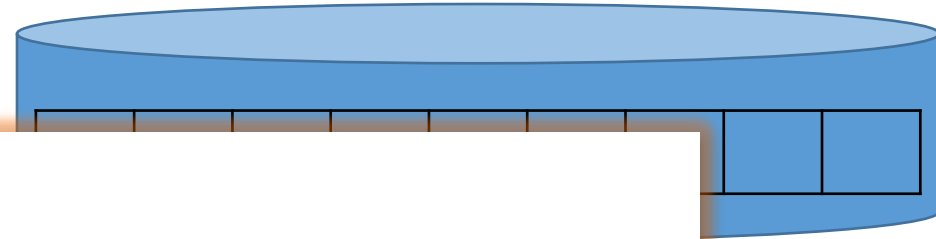
input3



И так далее...

Блоки, в которые вычитывается содержимое  
входных файлов, продолжают заполняться по  
мере их освобождения

DISK



OUTPUT



# Итак...

Использование внешней сортировки направлено на то, чтобы *уменьшить количество обращений ввода-вывода к диску*, а также выжать максимум из доступной оперативной памяти

# Итак... #2

1. Входной файл разбивается на блоки, для которых выполняется сортировка  
*Блоки должны быть как можно большего размера*

# Итак... #2

1. Входной файл разбивается на блоки, для которых выполняется сортировка  
*Блоки должны быть как можно большего размера*
2. Затем выполняется слияние предварительно отсортированных блоков и запись на диск