

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

5 2 3 4 1

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 4, j = 0$

2 5 3 4 1

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 4, j = 1$

2 3 5 4 1

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 4, j = 2$

2 3 4 5 1

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 4, j = 3$

2 3 4 1 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 3, j = 0$

2 3 4 1 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 3, j = 1$

2 3 4 1 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 3, j = 2$

2 3 1 4 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 2, j = 0$

2 3 1 4 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 2, j = 1$

2 1 3 4 5

Пузырьковая сортировка

```
void Sort(int[] array)
{
    for (int i=array.Length-1;i>0;i--)
        for (int j=1;j<=i;j++)
            if (array[j-1]>array[j])
            {
                var temp=array[j-1];
                array[j-1]=array[j];
                array[j]=temp;
            }
}
```

$i = 1, j = 0$

1 2 3 4 5

Слияние

↓		
1	2	6
3	4	5
↑		

Слияние

↓			
1	2	6	
3	4	5	1
↑			

Слияние

	↓		
1	2	6	
3	4	5	
↑			

1 2

Слияние

	↓		
1	2	6	
3	4	5	
↑			

1 2 3

Слияние

	↓		
1	2	6	
3	4	5	
	↑		

1 2 3 4

Слияние

	↓			
1	2	6		
3	4	5		
		↑		

1 2 3 4 5

Слияние

		↓	
1	2	6	
3	4	5	
		↑	

1 2 3 4 5 6

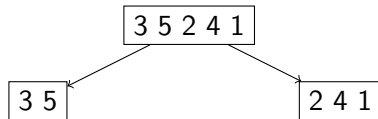
```
static void Merge(int[] array, int start, int middle, int end)
{
    var leftPtr = start;
    var rightPtr = middle + 1;
    var length = end - start + 1;
    for (int i = 0; i < length; i++)
        if ( rightPtr > end ||
            (leftPtr <= middle && array[leftPtr] < array[rightPtr]))
        {
            temporaryArray[i] = array[leftPtr];
            leftPtr++;
        }
        else
        {
            temporaryArray[i] = array[rightPtr];
            rightPtr++;
        }

    for (int i = 0; i < length; i++) array[i + start] = temporaryArray[i];
}
```

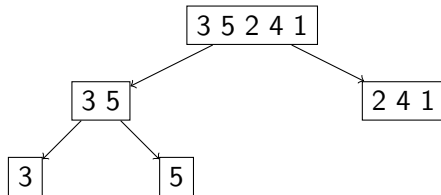
Сортировка слиянием

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

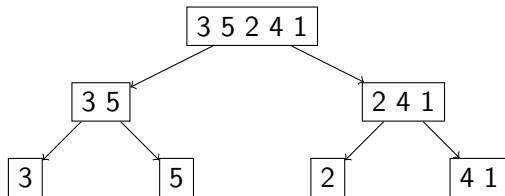
Сортировка слиянием



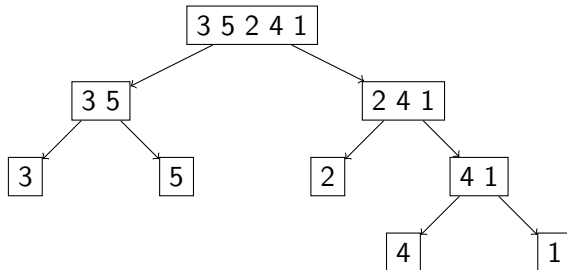
Сортировка слиянием



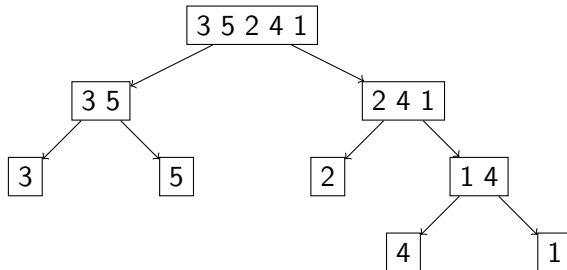
Сортировка слиянием



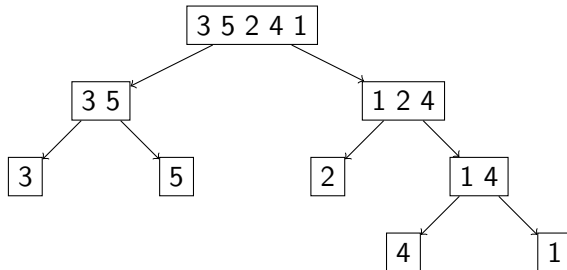
Сортировка слиянием



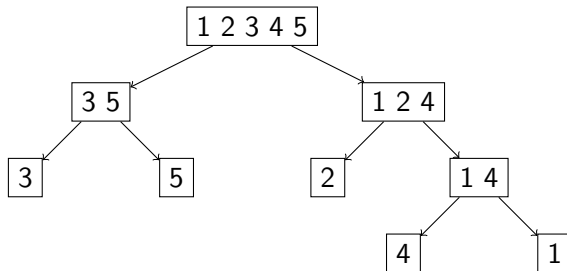
Сортировка слиянием



Сортировка слиянием



Сортировка слиянием



Сортировка слиянием

```
static void MergeSort(int[] array, int start, int end)
{
    if (start == end) return;
    var middle = (start + end) / 2;
    MergeSort(array, start, middle);
    MergeSort(array, middle + 1, end);
    Merge(array, start, middle, end);
}
```

Быстрая сортировка

5 1 4 2 3

Быстрая сортировка

5 1 4 2 3

Быстрая сортировка

5 1 4 2 3
↑

Быстрая сортировка

↓
5 1 4 2 3
↑

Быстрая сортировка

5 1 4 2 3
↑ ↓

Быстрая сортировка

1 5 4 2 3
↑ ↓

Быстрая сортировка

1 5 4 2 3

↓
↑

Быстрая сортировка

1 5 4 2 3

↑ ↓

Быстрая сортировка

1 5 4 2 3

↑ ↓

Быстрая сортировка

1 2 4 5 3

↑ ↓

Быстрая сортировка

1 2 4 5 3
↑

Быстрая сортировка

1 2 3 5 4
↑


```
static void HoareSort(int[] array, int start, int end)
{
    if (end == start) return;
    var pivot=array[end];
    var storeIndex=start;
    for (int i=start;i<=end-1;i++)
        if (array[i]<=pivot)
        {
            array.Switch(i,storeIndex);
            storeIndex++;
        }
    array.Switch(storeIndex,end);
    if (storeIndex > start) HoareSort(array, start, storeIndex - 1);
    if (storeIndex < end) HoareSort(array, storeIndex + 1, end);
}
```