

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
(assume that the call to mergeSort on the line above completed all it needed to do)

--	--	--	--	--	--	--	--	--	--

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
(assume that the call to mergeSort on the line above completed all it needed to do)

2	6	7	11	13	4	8	15	9	3
---	---	---	----	----	---	---	----	---	---

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
 (assume that the call to mergeSort on the line above
 completed all it needed to do)

2	6	7	11	13	4	8	15	9	3
---	---	---	----	----	---	---	----	---	---

What will above array look like at line marked B?
 (assume that the call to mergeSort on the line above
 completed all it needed to do)

--	--	--	--	--	--	--	--	--	--

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
  //A
  mergeSort(array, mid+1, last)
  //B
  merge( array, first, mid,
         mid+1, last)
  //C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
 (assume that the call to mergeSort on the line above
 completed all it needed to do)

2	6	7	11	13	4	8	15	9	3
---	---	---	----	----	---	---	----	---	---

What will above array look like at line marked B?
 (assume that the call to mergeSort on the line above
 completed all it needed to do)

2	6	7	11	13	3	4	8	9	15
---	---	---	----	----	---	---	---	---	----

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
(assume that the call to mergeSort on the line above completed all it needed to do)

2	6	7	11	13	4	8	15	9	3
---	---	---	----	----	---	---	----	---	---

What will above array look like at line marked B?
(assume that the call to mergeSort on the line above completed all it needed to do)

2	6	7	11	13	3	4	8	9	15
---	---	---	----	----	---	---	---	---	----

What will above array look like at line marked C?
(assume that the call to merge on the line above completed all it needed to do)

--	--	--	--	--	--	--	--	--	--

```
mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A?
(assume that the call to mergeSort on the line above completed all it needed to do)

2	6	7	11	13	4	8	15	9	3
---	---	---	----	----	---	---	----	---	---

What will above array look like at line marked B?
(assume that the call to mergeSort on the line above completed all it needed to do)

2	6	7	11	13	3	4	8	9	15
---	---	---	----	----	---	---	---	---	----

What will above array look like at line marked C?
(assume that the call to merge on the line above completed all it needed to do)

2	3	4	6	7	8	9	11	13	15
---	---	---	---	---	---	---	----	----	----

```
mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
        mid+1, last)
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

--	--	--	--	--	--	--	--	--	--

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```


13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked B when that line is reached for the very first time in nested recursive calls?

--	--	--	--	--	--	--	--	--	--

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked B when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked B when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very first time in nested recursive calls?

--	--	--	--	--	--	--	--	--	--

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked A when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked B when that line is reached for the very first time in nested recursive calls?

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very first time in nested recursive calls?

6	13	2	7	11	4	8	15	9	3
---	----	---	---	----	---	---	----	---	---

```
mergeSort( array , first, last )
    if first >= last
        return
    mid = (first+last)/2
    mergeSort(array, first, mid)
//A
    mergeSort(array, mid+1, last)
//B
    merge( array, first, mid,
           mid+1, last)
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?

6	13	2	7	11	4	8	15	9	3
---	----	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?

--	--	--	--	--	--	--	--	--	--

```

mergeSort( array , first, last )
    if first >= last
        return
    mid = (first+last)/2
    mergeSort(array, first, mid)
//A
    mergeSort(array, mid+1, last)
//B
    merge( array, first, mid,
           mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?

6	13	2	7	11	4	8	15	9	3
---	----	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?

2	6	13	7	11	4	8	15	9	3
---	---	----	---	----	---	---	----	---	---

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
        mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?

6	13	2	7	11	4	8	15	9	3
---	----	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?

2	6	13	7	11	4	8	15	9	3
---	---	----	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very THIRD time in nested recursive calls?

--	--	--	--	--	--	--	--	--	--

```

mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```


13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?

6	13	2	7	11	4	8	15	9	3
---	----	---	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?

2	6	13	7	11	4	8	15	9	3
---	---	----	---	----	---	---	----	---	---

What will above array look like at line marked C when that line is reached for the very THIRD time in nested recursive calls?

2	6	13	7	11	4	8	15	9	3
---	---	----	---	----	---	---	----	---	---

```

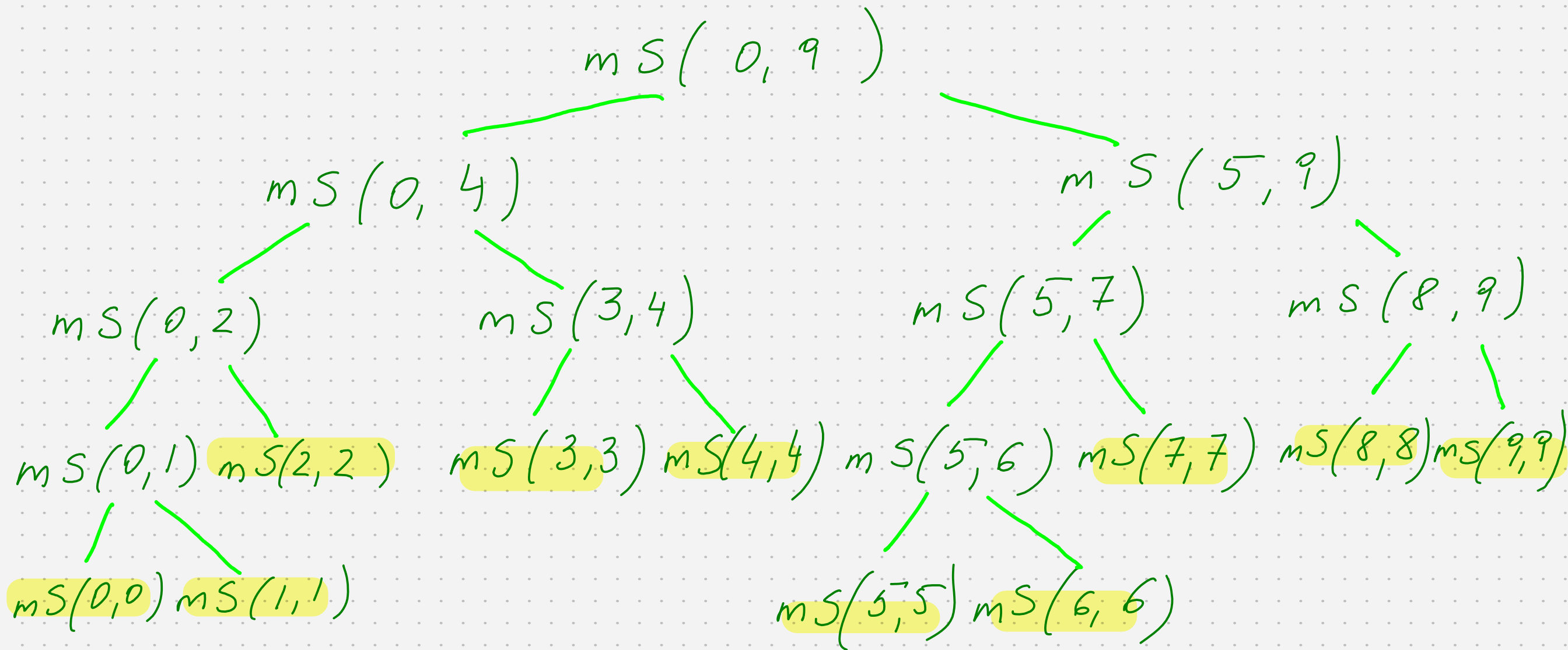
mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What is the total number of recursive calls made to mergeSort function in the process of sorting the array above?

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What is the total number of recursive calls made to mergeSort function in the process of sorting the array above?

19

What is the total number of calls made to merge function in the process of sorting the array above?

```
mergeSort( array , first, last )  
    if first >= last  
        return  
    mid = (first+last)/2  
    mergeSort(array, first, mid)  
//A  
    mergeSort(array, mid+1, last)  
//B  
    merge( array, first, mid,  
           mid+1, last)  
//C
```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What is the total number of recursive calls made to mergeSort function in the process of sorting the array above?

19

What is the total number of calls made to merge function in the process of sorting the array above?

9 (just count each pair of / \ in the recursion tree)

```

mergeSort( array , first, last )
    if first >= last
        return
    mid = (first+last)/2
    mergeSort(array, first, mid)
//A
    mergeSort(array, mid+1, last)
//B
    merge( array, first, mid,
           mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What is the total number of recursive calls made to mergeSort function in the process of sorting the array above?

19

What is the total number of calls made to merge function in the process of sorting the array above?

9 (just count each pair of / \ in the recursion tree)

What is the largest number of frames for the mergeSort function on a stack at any given time in the process of sorting the array above?

```

mergeSort( array , first, last )
    if first >= last
        return
    mid = (first+last)/2
    mergeSort(array, first, mid)
//A
    mergeSort(array, mid+1, last)
//B
    merge( array, first, mid,
           mid+1, last)
//C

```

13	6	2	7	11	4	8	15	9	3
----	---	---	---	----	---	---	----	---	---

What is the total number of recursive calls made to mergeSort function in the process of sorting the array above?

19

What is the total number of calls made to merge function in the process of sorting the array above?

9 (just count each pair of / \ in the recursion tree)

What is the largest number of frames for the mergeSort function on a stack at any given time in the process of sorting the array above?

5 (just look at the "depth" of the recursion tree)

```
mergeSort( array , first, last )
  if first >= last
    return
  mid = (first+last)/2
  mergeSort(array, first, mid)
//A
  mergeSort(array, mid+1, last)
//B
  merge( array, first, mid,
         mid+1, last)
//C
```