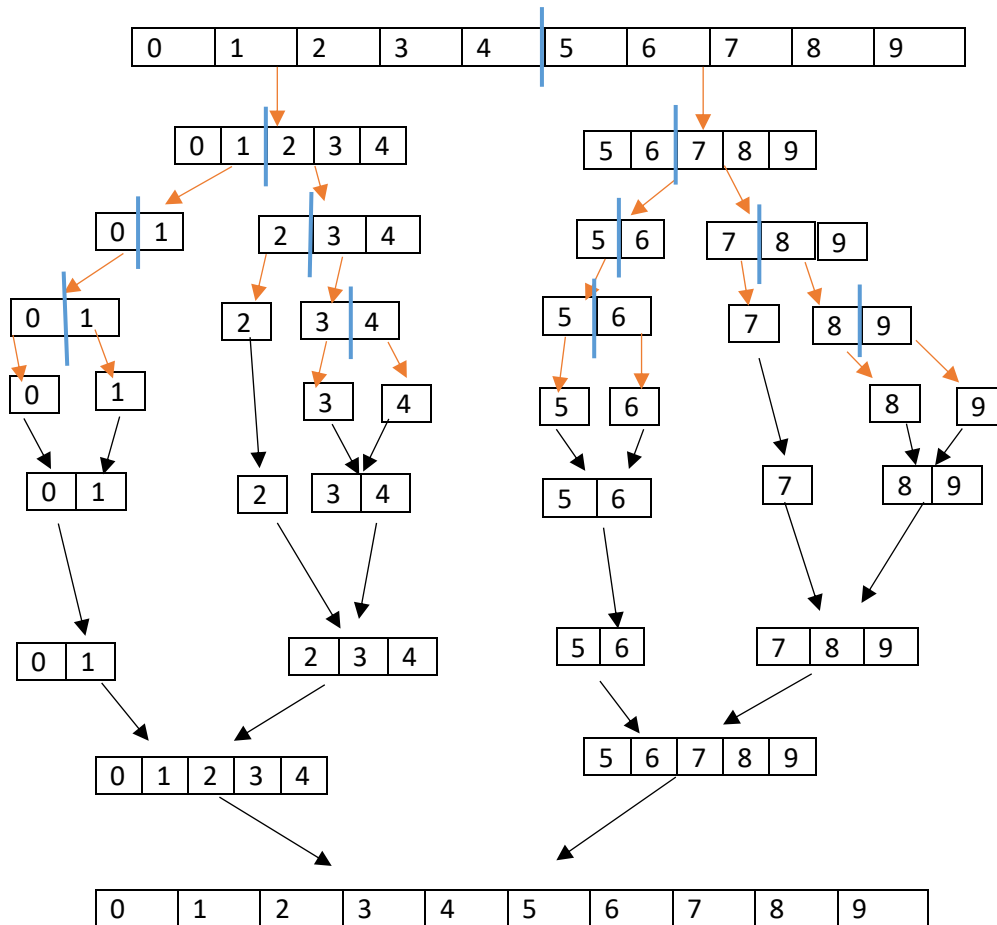


Merge Sort

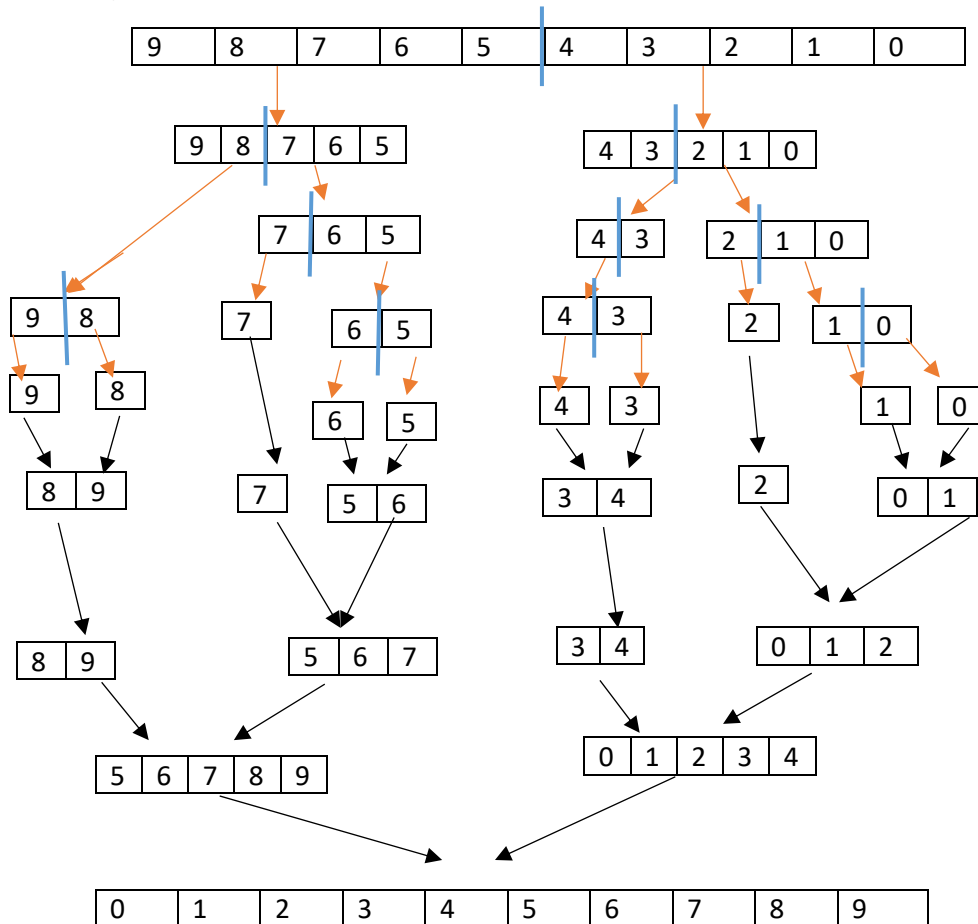
A)



On the upper side , The stages are written step by step.I show the deviding by using blue line, the showing arrays by using orane arrow and I show the merging by using black arrow. At the first array , We devide this to two array. Then We do this until we get two arrays or one array.. After that we start to merge arrays by comparing so that they will be ordered . In this example , I devide all parts then I compare elements. First 0 and 1 it is okay then merge them (swapping elemets if it needs [0,1]).Then 3 and 4, They are okay also then I merge them (swapping elemets if it needs [3,4]). After that I look [2] and and [3,4], I compare 2 and 3 then first element of array is 2 becouse of being small than 4, afterthat the second element of the array is 3 the last element of the array is 4.So The array is [2,3,4].Also I compare [0,1] and [2,3,4]. Then I compare 0 and 2 ,becouse of being small 0 than 2, the first element of our array is 0 (no changing) then I compare 2 with 1 (no changing) then the second element is 1.There is no index to compare with 2. So our array is renewed. And left side array is [0,1,2,3,4]. Then I do same things on the right of midle in the first array. So the last step is comparing ordered left side and ordered right side one by one.After that our array is renewed and all ordered.These are steps my merge sort algorithm.

Total comparision = 15 and Total displacement =0.

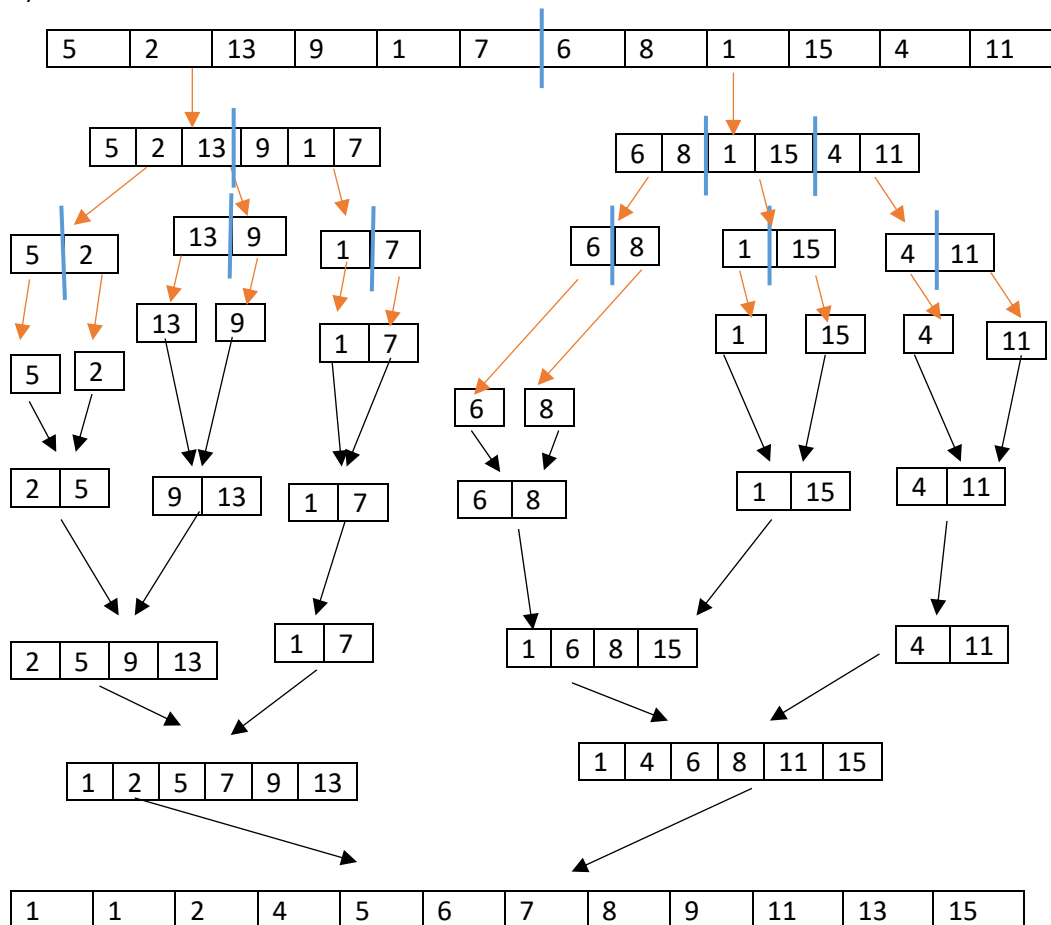
B)



On the upper side , The stages are written step by step.I show the deviding by using blue line, the showing arrays by using orane arrow and I show the merging by using black arrow. At the first array , We devide this to two array. Then We do this until we get two arrays or one array. After that we start to merge arrays by comparing so that they will be ordered . In this example , I devide all parts then I compare elements. First 9 and 8 it is not okay then merge them (swapping elemets if it needs [8,9]).Then 6 and 5, They are not okay also then I merge them (swapping elemets if it needs [5,6]). After that I look [7] and [5,6] , I compare 7 and 5 then first element of array is 5 because of being small than 7 then I compare 7 and 6 then the second element is 6 becauseof being small than 7 .There is no index to compare with 7.So The array is [5,6,7].Also I compare [8,9] and [5,6,7]. Then I compare 8 and 5 ,because of being small 5 than 8, the first element of our array is 5, then I compare 8 with 6 then the second element is 6 , then I compare 8 with 7 then the third element is 7.There is no index to compare with 8. So our array is renewed. And left side array is [5,6,7,8,9]. Then I do same things on the right of middle in the first array. So the last step is comparing ordered left side and ordered right side one by one.After that our array is renewed and all ordered.These are steps my merge sort algorithm.

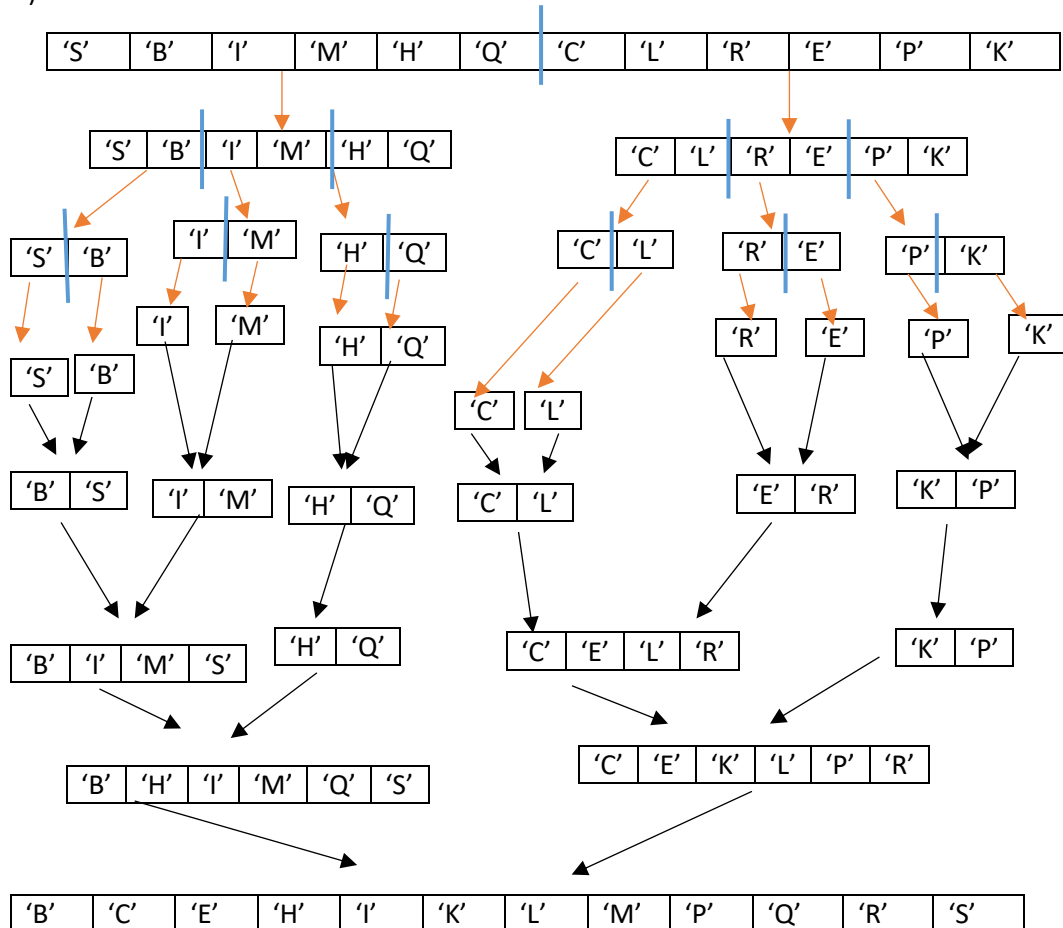
Total comparision = 19 and Total displacement = 13

c)



On the upper side , The stages are written step by step.I show the deviding by using blue line, the showing arrays by using orane arrow and I show the merging by using black arrow. At the first array , We devide this to two array. Then We do this until we get two element arrays or one element arrays. After that we start to merge arrays by comparing so that they will be ordered . In this example , I devide all parts then I compare elements. First 5 and 2 it is not okay then merge them (swapping elemets if it needs [2,5]).Then 13 and 9, They are not okay also then I merge them (swapping elemets if it needs [9,13]). Then 1 and 7, They are okay (no changing) then I merge them (swapping elemets if it needs [1,7]). After that I look [2,5] and [9,13], I compare 2 and 9 then first element of array is 2 because of being small than 9 , then I compare 9 and 5 then the second element of array is 6 because of being small than 9. After that There is no element to compare 9. So The array is [2,5,9,13].Also I compare [2,5,9,13] and [1,7]. Then I compare 2 and 1 ,because of being small 1 than 2, the first element of our array is 1 (swapping), then I compare 2 with 7 then the second element is 2, then I compare 7 with 5 then the third element is 5. Then I compare 7 and 9 , then the third element is 7. There is no index to compare with 9. So our array is renewed. And left side array is [1,2,5,7,9,13]. Then I do same things on the right of midle in the first array. So the last step is comparing ordered left side and ordered right side one by one.After that our array is renewed and all ordered.These are steps my merge sort algorithm. Total comparision = 18 and total displacemnt=10

D)



On the upper side , The stages are written step by step.I show the deviding by using blue line, the showing arrays by using orane arrow and I show the merging by using black arrow. At the first array , We devide this to two array. Then We do this until we get two element arrays or one element arrays. After that we start to merge arrays by comparing so that they will be ordered . In this example , I devide all parts then I compare elements. First 'S' and 'B' it is not okay then merge them (swapping elemets if it needs ['B','S']).Then 'I' and 'M'. They are okay then I merge them (swapping elemets if it needs ['I','M']).).Then 'H' and 'Q', They are okay (no changing) then I merge them (swapping elemets if it needs ['H','Q']). After that I look ['B','S'] and ['I','M'], I compare 'B' and 'I' then first element of array is 'B' because of being small than 'I' , then I compare 'I' and 'S' then the second element of array is 'I' because of being small than 'S'. After that then I compare 'S' and 'M' then the third element of array is 'M' because of being small than 'S'. So The array is ['B','I','M','S'].Also I compare ['B','I','M','S'] and ['H','Q']. Then I compare 'B' and 'H',because of being small than 'H', the first element of our array is 'B' (no swapping), then I compare 'H' with 'I' then the second element is 'H', then I compare 'I' with 'Q' then the third element is 'I'. Then I compare 'Q' and 'M' , then the forth element is 'M'.Then I compare 'Q' and 'S' , then the fifth element is 'Q'.There is no index to compare with 'S'. So our array is renewed. And left side array is ['B','H','I','M','Q','S']. Then I do same

things on the right of middle in the first array. So the last step is comparing ordered left side and ordered right side one by one. After that our array is renewed and all ordered. These are steps of my merge sort algorithm. Total comparison = 16 and total displacement = 15

Quick Sort

A)

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



i= -1

J=0

Pivot

Compare 9 and 0 then $0 < 9$ i++ j++

i= 0

J=1

Compare 9 and 1 then $1 < 9$ i++ j++

i= 1

J=2

Compare 9 and 2 then $2 < 9$ i++ j++

i= 2

J=3

Compare 9 and 3 then $3 < 9$ i++ j++

i= 3

J=4

Compare 9 and 4 then $4 < 9$ i++ j++

i= 4

J=5

Compare 9 and 5 then $5 < 9$ i++ j++

i= 5

J=6

Compare 9 and 6 then $6 < 9$ $i++$ $j++$

$i = 6$
 $j = 7$

Compare 9 and 7 then $7 < 9$ $i++$ $j++$

$i = 7$
 $j = 8$

Compare 9 and 8 then $8 < 9$ $i++$ $j++$

$i = 8$
 $j = 9$

When j is equal pivot index then pivot displace $i+1$ th element but $i+1$ is 9 then pivot is now 8 it goes until $i > j$

Total comparison : $9+8+7+6+5+4+3+2+1 = 9*10/2 = 45$

Total displacement : 0

B)

9	8	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---	---

$i = -1$
 $j = 0$

Pivot

Compare 9 and 0 $9 > 0$ $j++$

$i = -1$
 $j = 1$

Compare 8 and 0 $8 > 0$ $j++$

$i = -1$
 $j = 2$

Compare 7 and 0 $7 > 0$ j++

i= -1
J=3

Compare 6 and 0 $6 > 0$ j++

i= -1
J=4

Compare 5 and 0 $5 > 0$ j++

i= -1
J=5

Compare 4 and 0 $4 > 0$ j++

i= -1
J=6

Compare 3 and 0 $3 > 0$ j++

i= -1
J=7

Compare 2 and 0 $2 > 0$ j++

i= -1
J=8

Compare 1 and 0 $1 > 0$ j++

i= -1
J=9

Then displace 0 and i+1 th element

0	8	7	6	5	4	3	2	1	9
---	---	---	---	---	---	---	---	---	---

8	7	6	5	4	3	2	1	9
---	---	---	---	---	---	---	---	---



Pivot

i= -1

J=0

Compare 8 and 9 , $8 < 9$ i++ j++

i= 0

J=1

Compare 7 and 9 , $7 < 9$ i++ j++

i= 1

J=2

Compare 6 and 9 , $6 < 9$ i++ j++

i= 3

J=4

Compare 5 and 9 , $5 < 9$ i++ j++

i= 4

J=5

Compare 4 and 9 , $4 < 9$ i++ j++

i= 5

J=6

Compare 3 and 9 , $3 < 9$ i++ j++

i= 6

J=7

Compare 2 and 9 , $2 < 9$ i++ j++

i= 7

J=8

Compare 1 and 9 , $1 < 9$ $i++$ $j++$

$i = 8$

$J = 9$

J is now pivot then pivot is displaced with $i+1$ th element So it is itself then pivot is 8 , so it continues like this until $i > j$

Total comparison is : 45

Total displacement is : 9

C)

5	2	13	9	1	7	6	8	1	15	4	11
---	---	----	---	---	---	---	---	---	----	---	----

Pivot

$i = -1$

$J = 0$

Compare 5 and 11 $j++$

$i = -1$

$J = 1$

Compare 2 and 11 , $2 < 11$ $j++$

$i = -1$

$J = 2$

Compare 13 and 11 , $13 > 11$ $i++$ $j++$

$i = 0$

$J = 3$

Compare 9 and 11 , $9 < 11$ $j++$

$i = 0$

$J = 4$

Compare 1 and 11 ,1<11 j++

i= 0

J=5

Mustafa Tokgöz 171044077