

# Ch5: Merge Sort

305233, 305234

Algorithm Analysis and Design

Jiraporn Pooksook  
Naresuan University

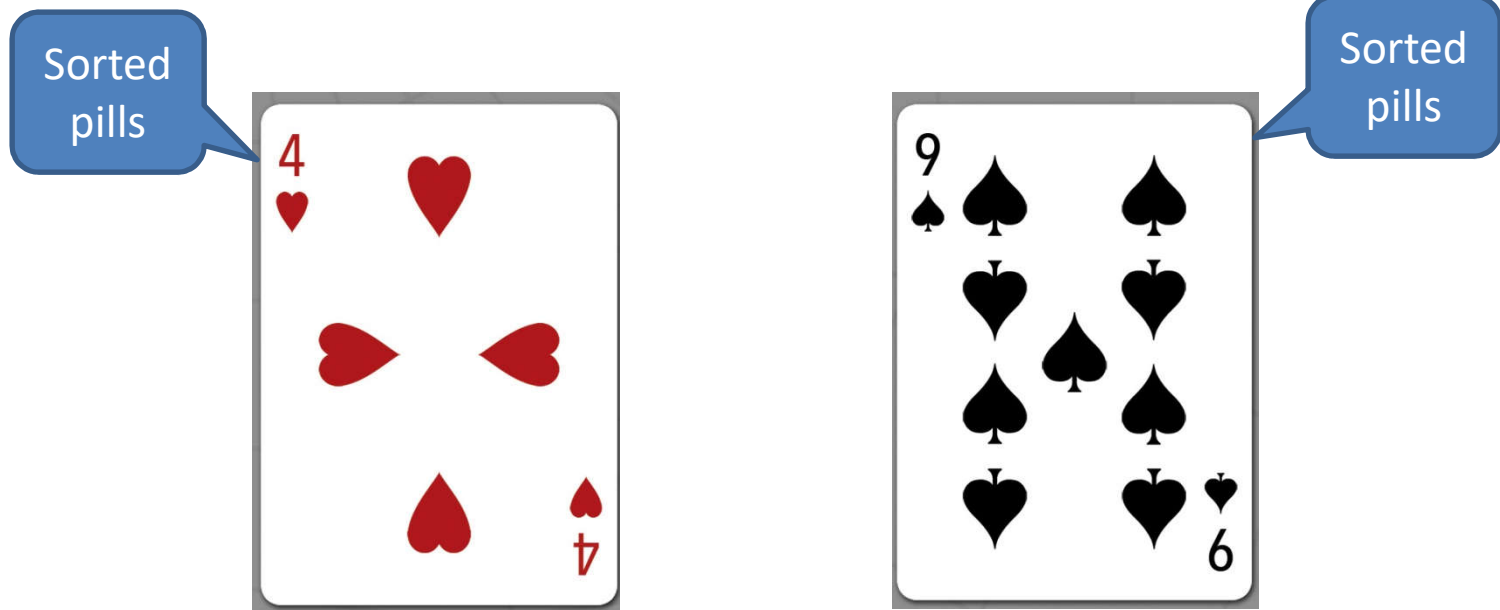
# Merge Sort vs Cards



This photo is taken from <https://www.pleasanttimes.com/images/2card-box-teak-2.jpg>

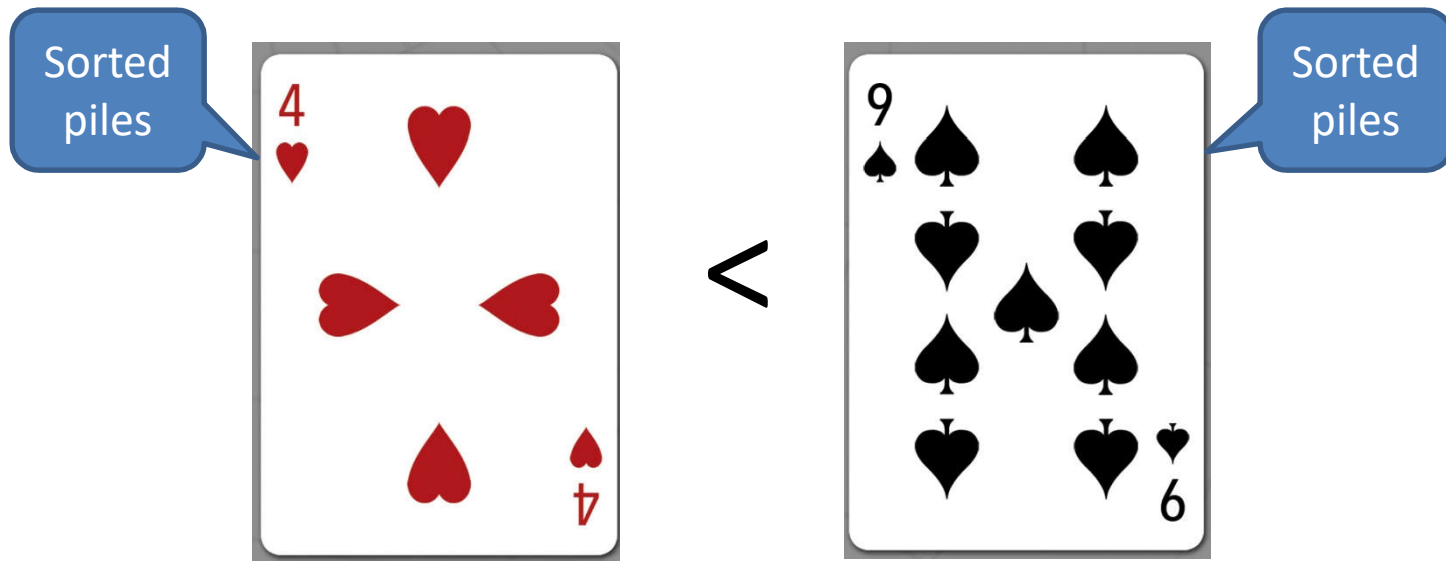
# Merge Sort vs Cards

- We have two sorted piles of a card deck.



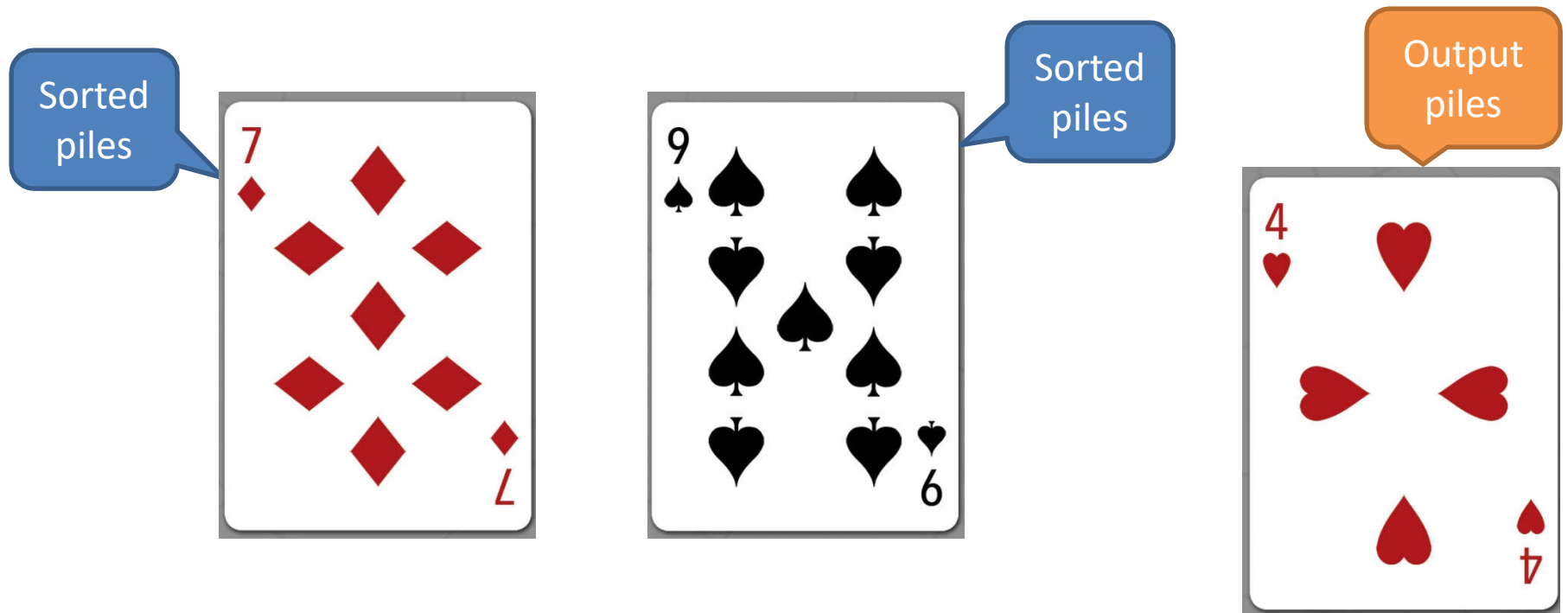
# Merge Sort vs Cards

- We compare the top card of both piles and put the smaller card on another pile.

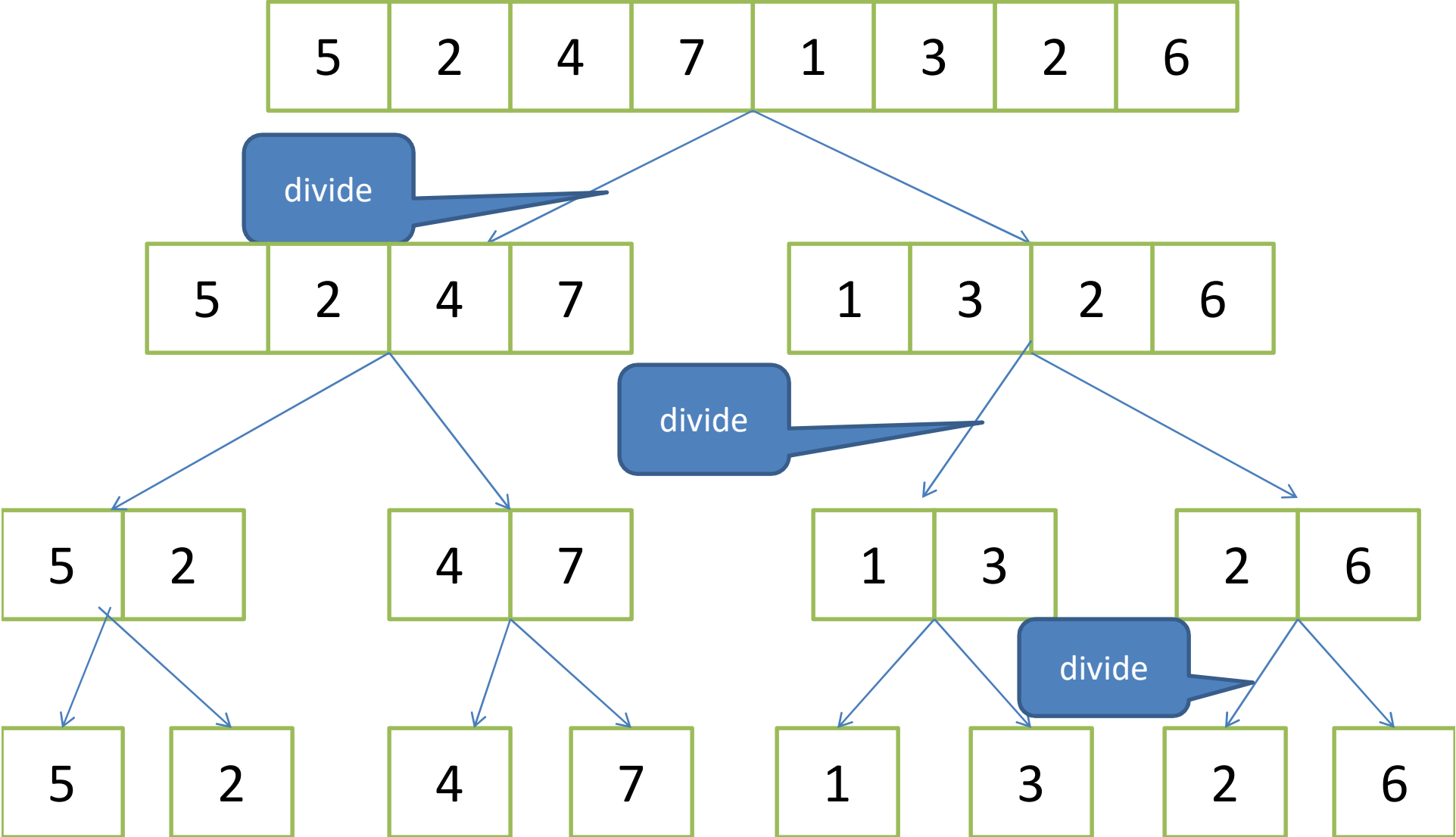


# Merge Sort vs Cards

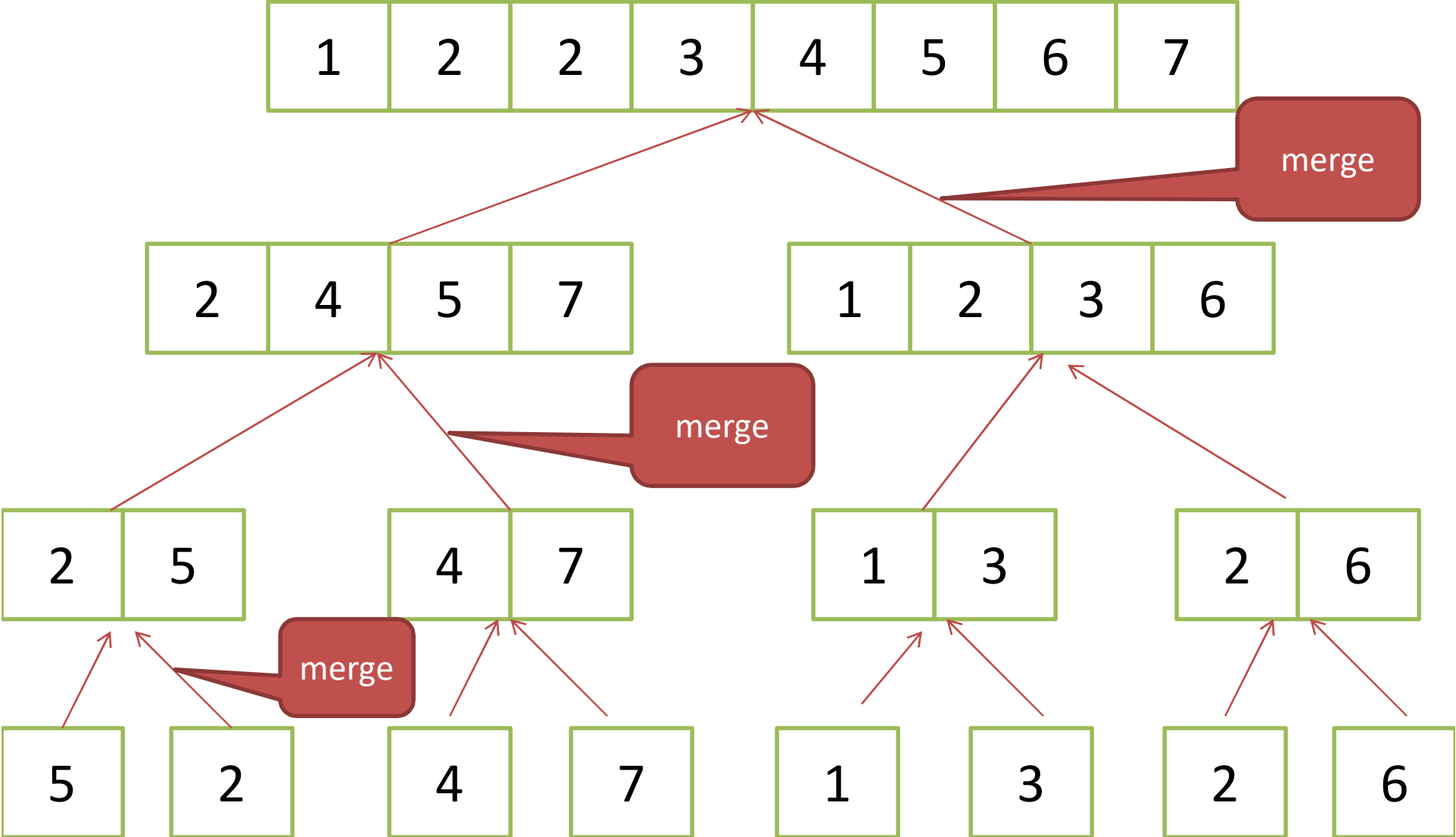
- We compare the top card of both piles and put the smaller card on another pile. Then we compare the next cards.



# Merge Sort



# Merge Sort



# Merge Sort

- Divide the  $n$  element arrays to be two arrays of size  $n/2$
- Sort the two divided arrays using merge sort recursively
- Merge the two sorted arrays to produce the sorted answer



# Pseudocode: merge(A,p,q,r)

```
N1 = q - p + 1
N2 = r - q
Create arrays L[1..N1+1] and R[1...N2+1]
For i = 1 to N1
    do L[i] = A[p + i - 1]
For j = 1 to N2
    do R[j] = A[q + j]
L[N1+1] = ∞
R[N2+1] = ∞
i = 1
j = 1
For k=p to r
    do if L[ i ] <= R[ j ]
        then A[k] = L[ i ]
            i = i+1
        else A[k] = R[ j ]
            j = j+1
```

# Practice: Merge(A,p,q,r)

K

...	2	4	5	7	1	2	3	6	...
-----	---	---	---	---	---	---	---	---	-----

L

2	4	5	7	$\infty$
---	---	---	---	----------

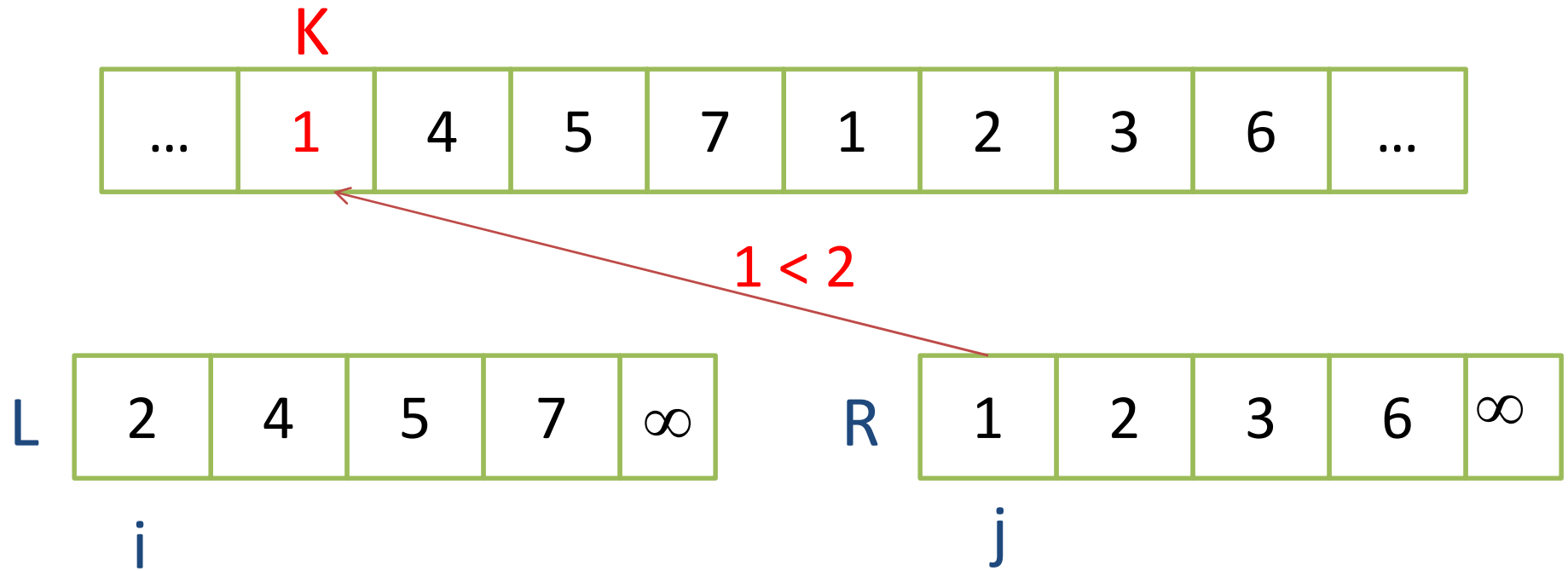
i

R

1	2	3	6	$\infty$
---	---	---	---	----------

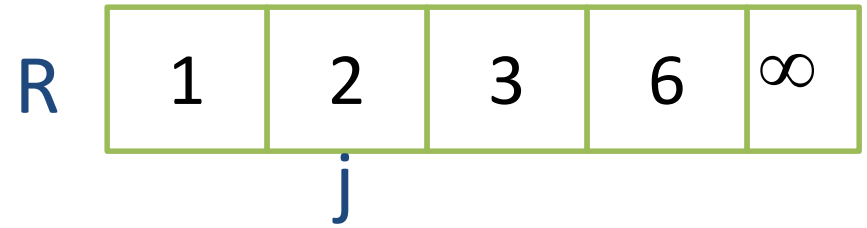
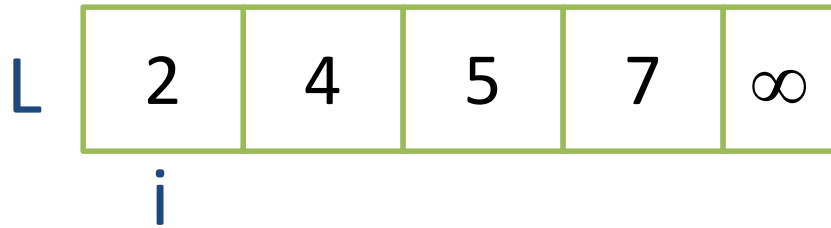
j

# Practice: Merge(A,p,q,r)



# Practice: Merge(A,p,q,r)

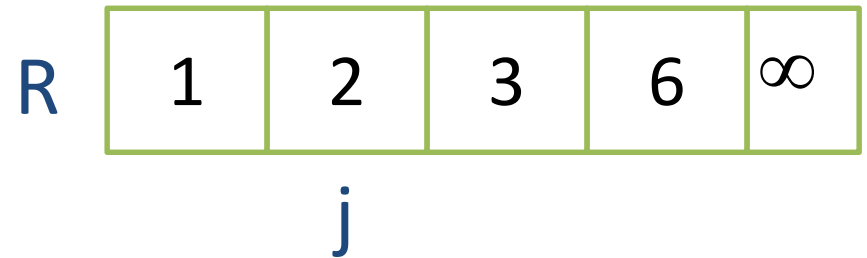
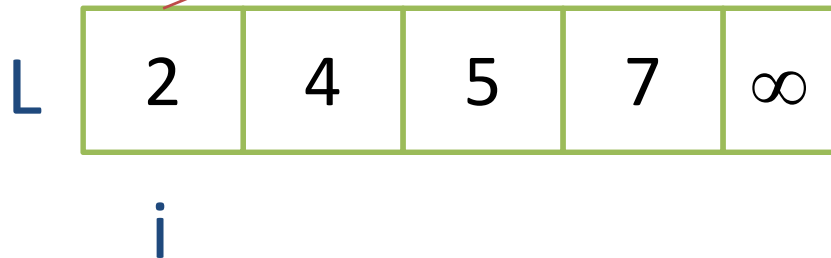
K



K

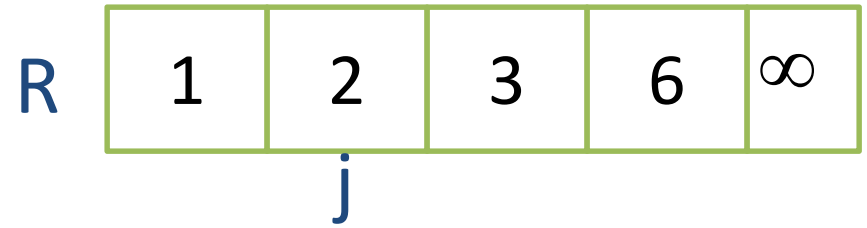


$2 \leq 2$

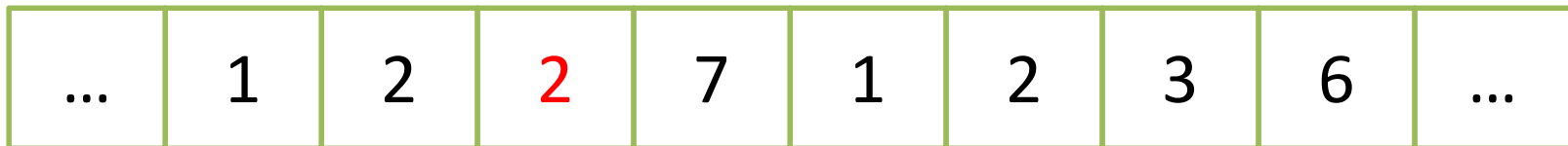


# Practice: Merge(A,p,q,r)

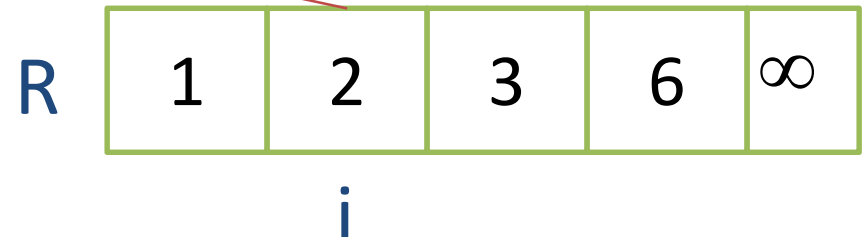
K



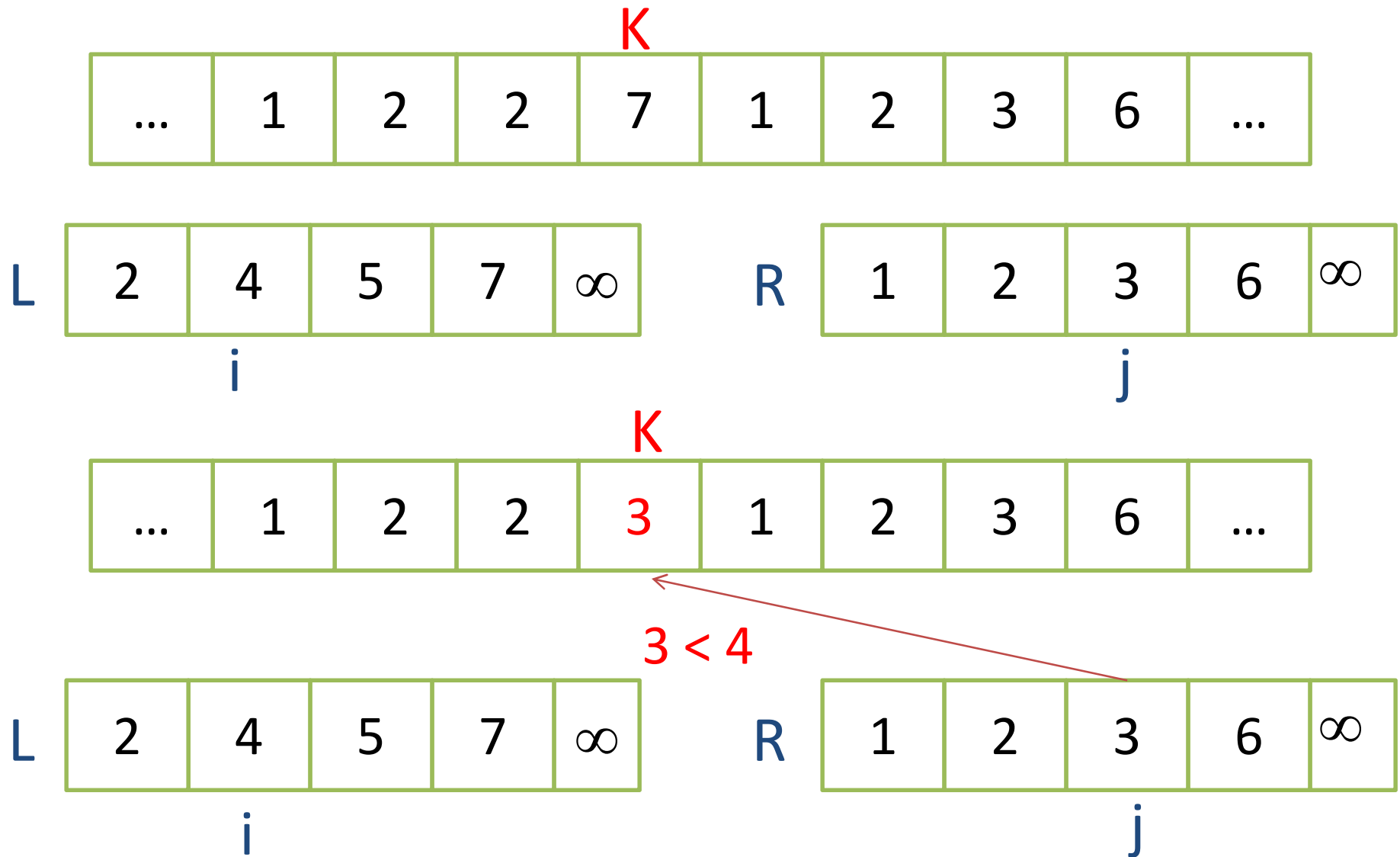
K



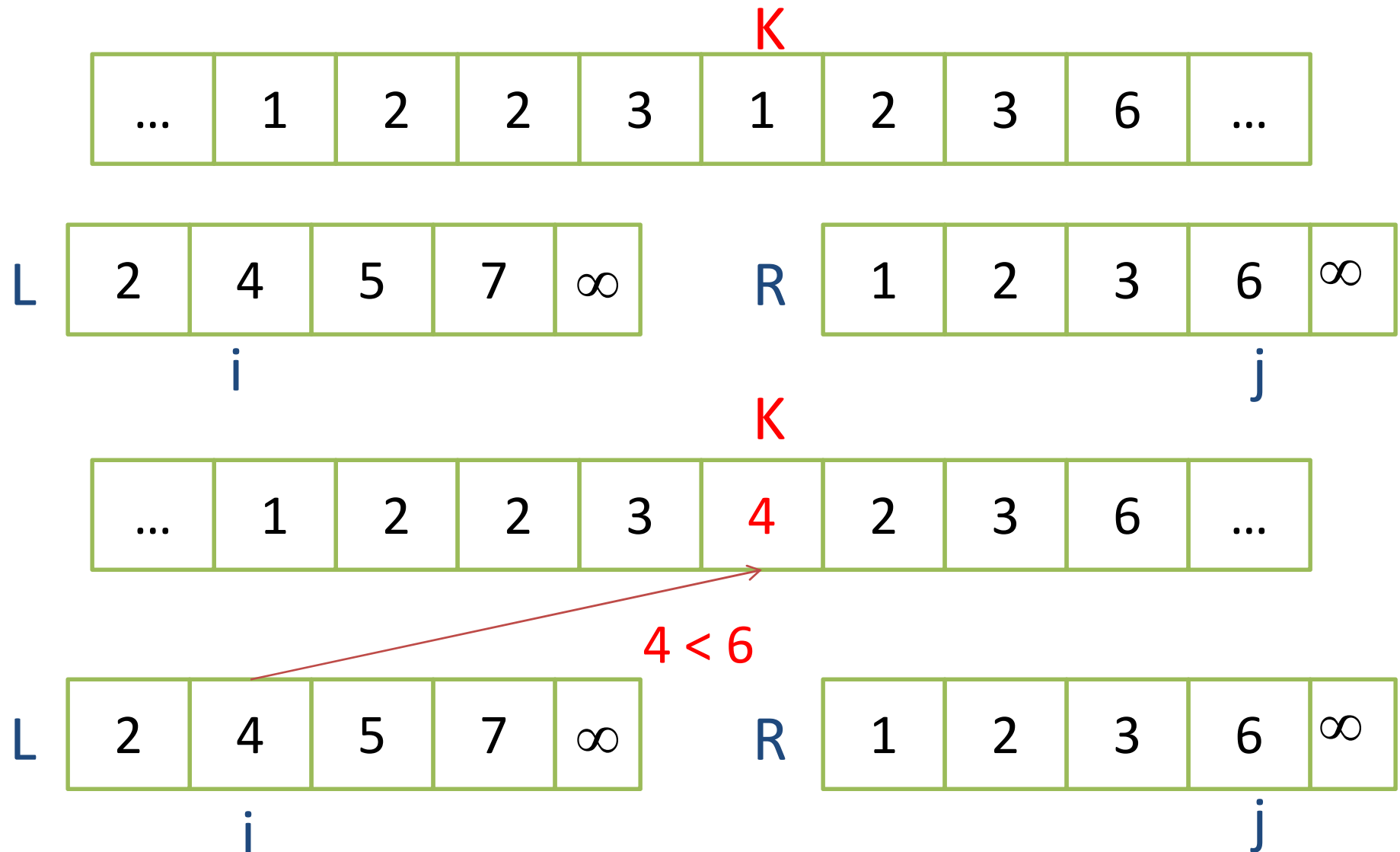
$2 < 4$



# Practice: Merge(A,p,q,r)



# Practice: Merge(A,p,q,r)



# Practice: Merge(A,p,q,r)

K



L



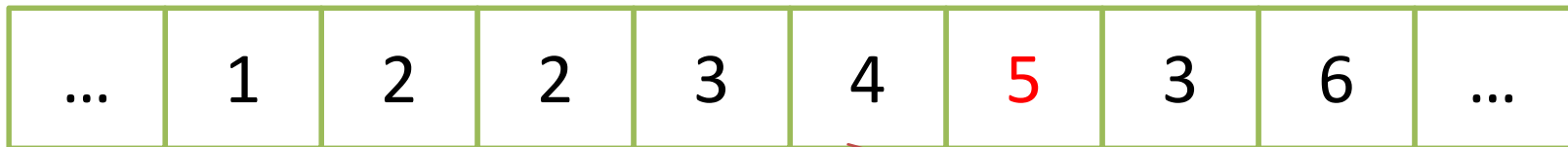
i

R



j

K



5 < 6

L



i

R



j



# Practice: Merge(A,p,q,r)

K



L



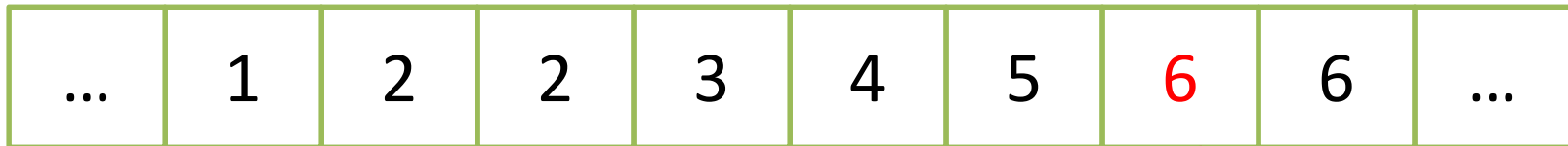
i

R



j

K



6 < 7

L



i

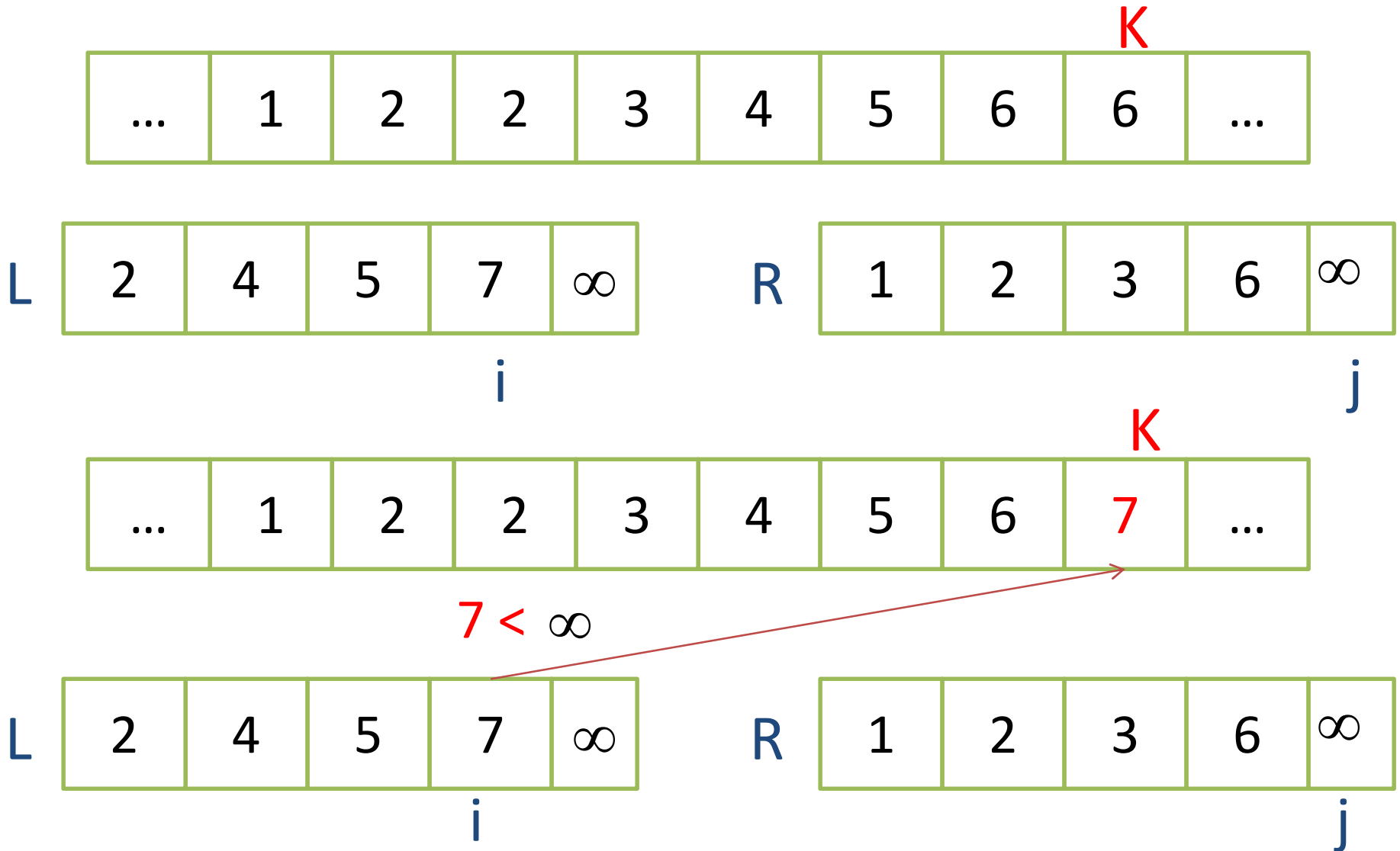
R



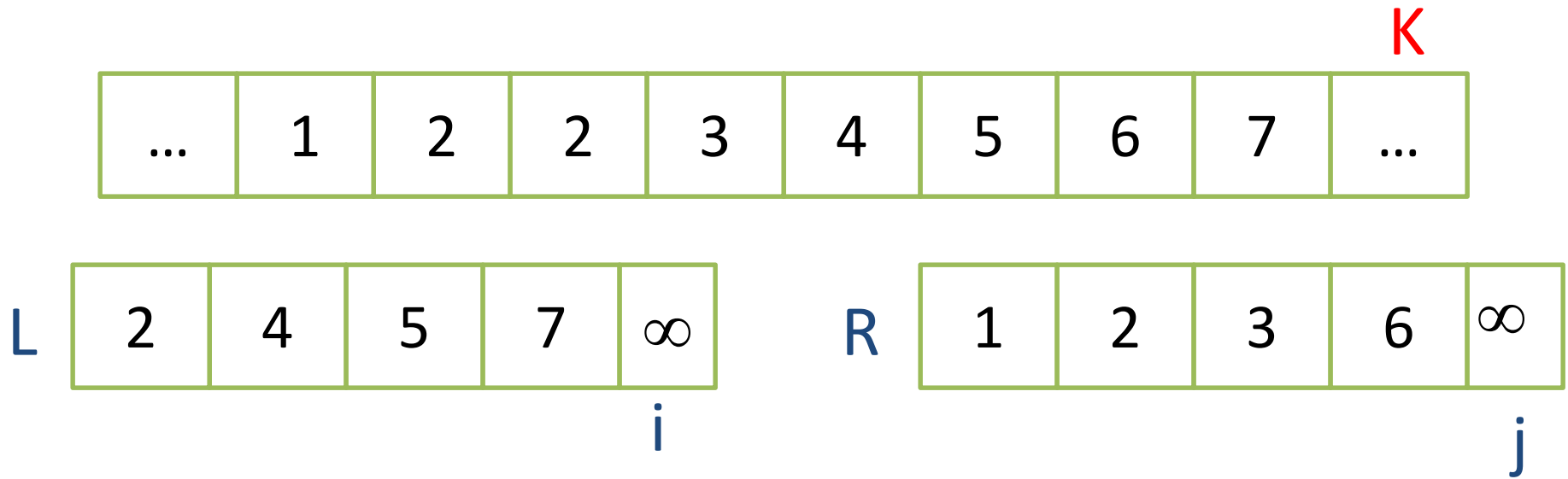
j



# Practice: Merge(A,p,q,r)



# Practice: Merge(A,p,q,r)



# Pseudocode: merge-sort (A,p,r)

If  $p < r$

    then  $q = \lfloor (p+r)/2 \rfloor$

        merge-sort(A, p, q)

        merge-sort(A, q+1, r)

        merge(A, p, q, r)

# Practice: merge-sort (A,1,8)

If  $p < r$

then  $q = \lfloor (p+r)/2 \rfloor$

round 1 ,  $q = 4$

round 2 ,  $q = 2$

merge-sort(A, p, q)

round 1 , merge-sort(A,1,4)

merge-sort(A, q+1, r)

round 1 , merge-sort(A,5,8)

merge(A, p, q, r)

round 1 , merge(A,1,4,8)

# Practice: merge-sort (A,1,4)

If  $p < r$

then  $q = \lfloor (p+r)/2 \rfloor$

round 2 ,  $q = 2$

merge-sort(A, p, q)

**round 1 , merge-sort(A,1,4)**

round 2 , merge-sort(A,1,2)

merge-sort(A, q+1, r)

**round 1 , merge-sort(A,5,8)**

round 2 , merge-sort(A,3,4)

merge(A, p, q, r)

**round 1 , merge(A,1,4,8)**

round 2 , merge(A,1,2,4)

# Practice: merge-sort (A,5,8)

If  $p < r$

then  $q = \lfloor (p+r)/2 \rfloor$

round 2 ,  $q = 2$

merge-sort(A, p, q)

**round 1 , merge-sort(A,1,4)**

round 3 , merge-sort(A,5,6)

merge-sort(A, q+1, r)

**round 1 , merge-sort(A,5,8)**

round 3 , merge-sort(A,7,8)

merge(A, p, q, r)

**round 1 , merge(A,1,4,8)**

round 3 , merge(A,5,6,8)

# Practice: merge-sort (A,1,2)

If  $p < r$

then  $q = \lfloor (p+r)/2 \rfloor$

round 3 ,  $q = 1$

merge-sort(A, p, q)

round 1 , merge-sort(A,1,4)

round 2 , merge-sort(A,1,2)

round 3 , merge-sort(A,1,1)

merge-sort(A, q+1, r)

round 1 , merge-sort(A,5,8)

round 2 , merge-sort(A,3,4)

round 3 , merge-sort(A,2,2)

merge(A, p, q, r)

round 1 , merge(A,1,4,8)

round 2 , merge(A,1,2,4)

round 3 , merge(A,1,1,2)



# Practice: merge-sort (A,3,4)

If  $p < r$

then  $q = \lfloor (p+r)/2 \rfloor$

round 3 ,  $q = 3$

merge-sort(A, p, q)

round 1 , merge-sort(A,1,4)

round 2 , merge-sort(A,1,2)

round 3 , merge-sort(A,3,3)

merge-sort(A, q+1, r)

round 1 , merge-sort(A,5,8)

round 2 , merge-sort(A,3,4)

round 3 , merge-sort(A,4,4)

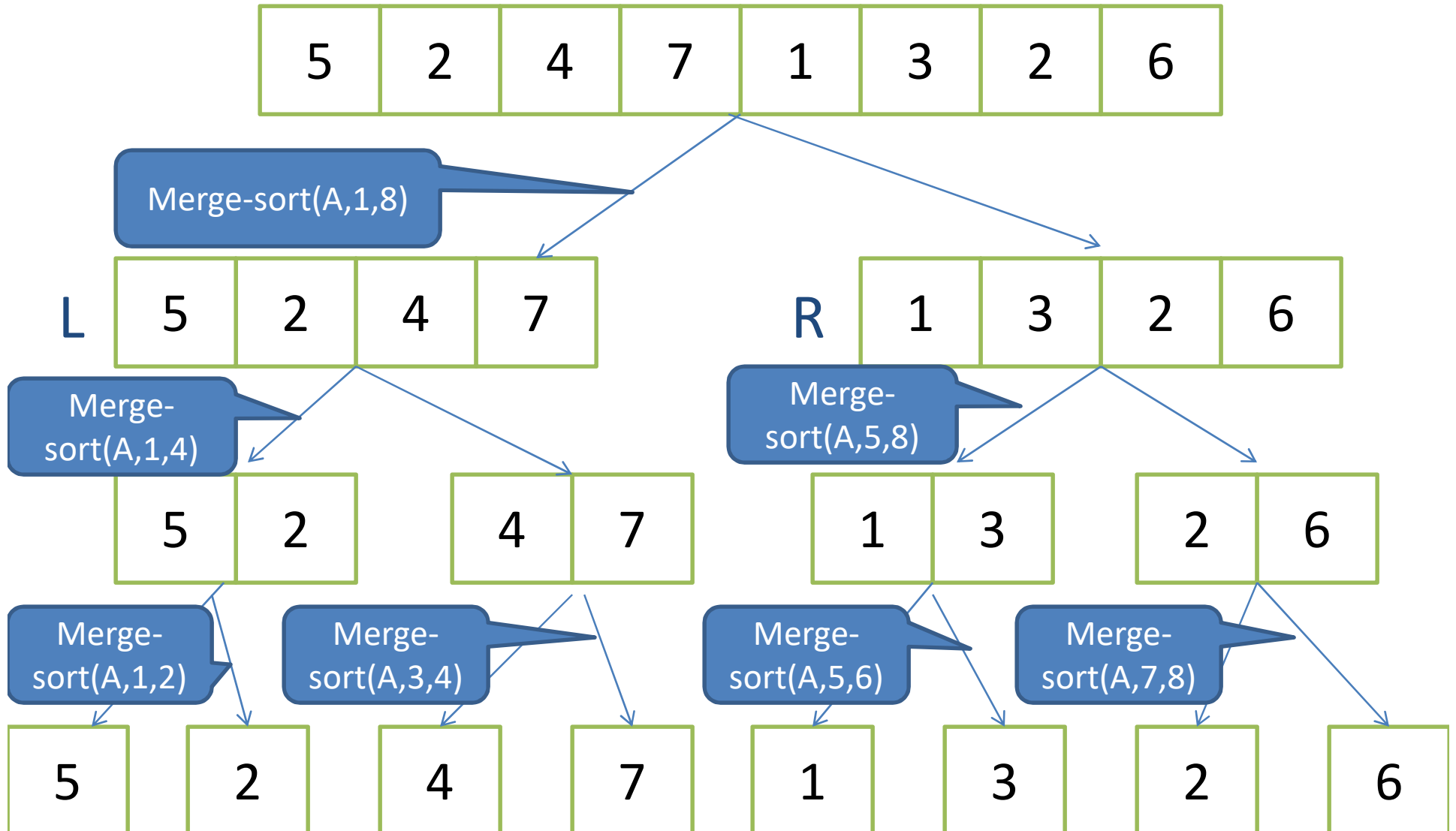
merge(A, p, q, r)

round 1 , merge(A,1,4,8)

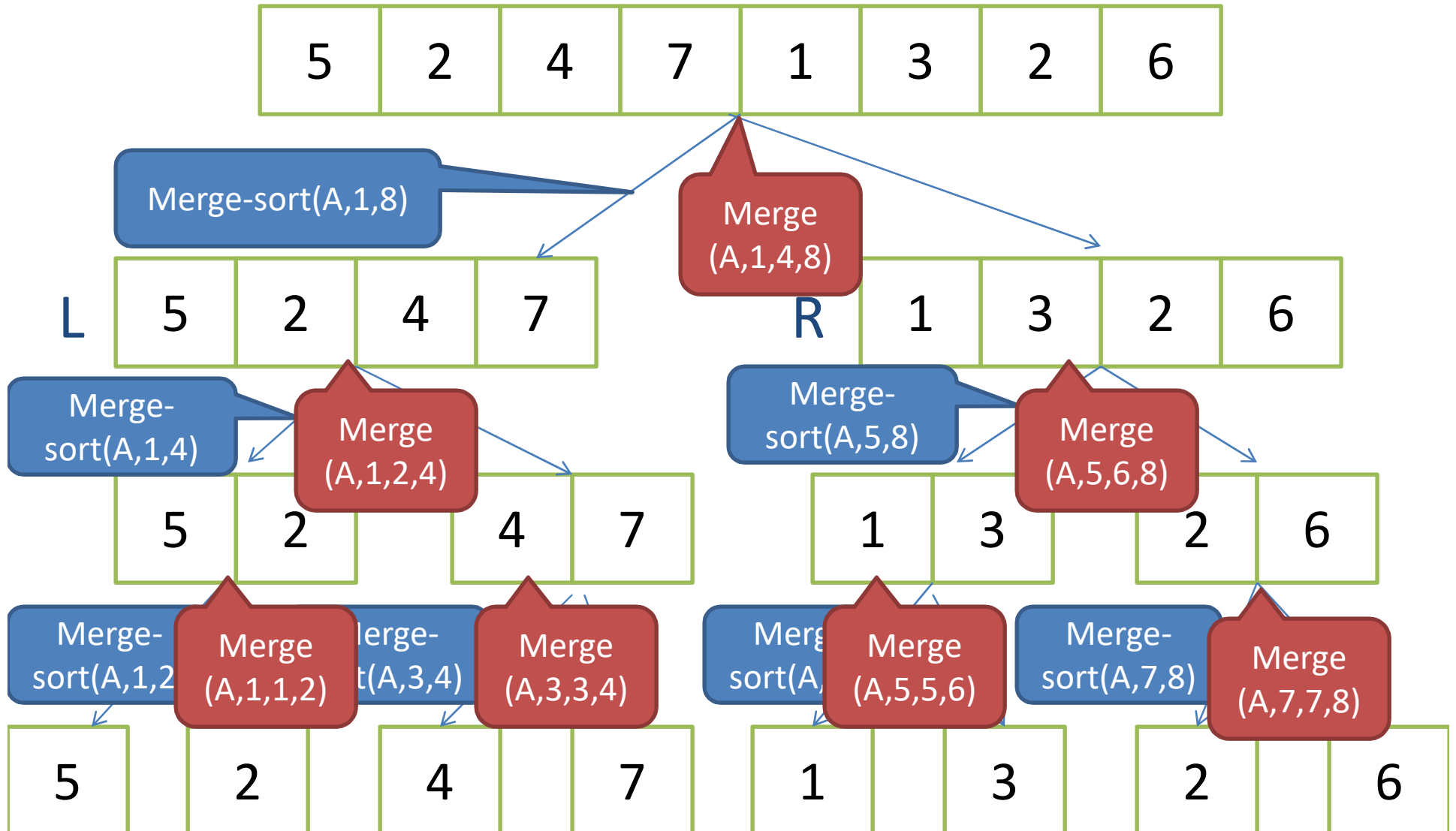
round 2 , merge(A,1,2,4)

round 3 , merge(A,3,3,4)

# Practice: Merge-sort(A,p,r)



# Practice: Merge-sort(A,p,r)



# Exercise merge-sort

Input = [9,5,7,4,2,8]