

# Phân tích và Thiết kế THUẬT TOÁN

Nguyễn Mậu Uyên

[uyennm@mta.edu.vn](mailto:uyennm@mta.edu.vn)

Web: [fit.mta.edu.vn/~uyennm](http://fit.mta.edu.vn/~uyennm)

# Bài 4 - Thiết kế thuật toán Chia để trị - Divide&Conquer

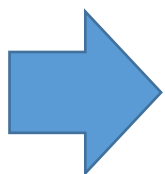
PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN

# NỘI DUNG

- I. Giới thiệu
- II. Lược đồ chung
- III. Bài toán áp dụng
- IV. Bài tập

# I. Giới thiệu

- Là một phương pháp được áp dụng rộng rãi
- Ý tưởng chung là phân rã bài toán thành bài toán nhỏ hơn “độc lập” với nhau.
- Giải các bài toán con theo cùng 1 cách thức
- “Tổng hợp” lời các bài toán con để có được kết quả bài toán ban đầu.



Tư tưởng chung của cách tiếp cận **Chia để trị**

## II. Lược đồ chung

### **Chia:**

- Bằng cách nào đó chia tập hợp các đối tượng của bài toán thành bài toán con “độc lập”
- Tiếp tục chia các bài toán con cho đến khi có thể giải trực tiếp (không cần, hoặc không thể chia nhỏ nữa)

### **Trị:**

- Trên các bài toán con thực hiện cùng một cách thức: Chia nhỏ nếu cần hoặc giải trực tiếp

### **Tổng hợp:**

- Khi mỗi bài toán con được giải, tổng hợp để có kết quả bài toán ban đầu.

## II. Lược đồ chung

Nếu gọi  $D\&C(\mathcal{R})$  - Với  $\mathcal{R}$  là miền dữ liệu  
là hàm thể hiện cách giải bài  
toán theo phương pháp chia để trị thì ta có thể viết :

```
void D&C( $\mathcal{R}$ )
{
    If ( $\mathcal{R}$  đủ nhỏ)
        giải bài toán;
    Else
    {
        Chia  $\mathcal{R}$  thành  $\mathcal{R}_1, \dots, \mathcal{R}_m$  ;
        for ( $i = 1; i \leq m; i++$ )
            D&C( $\mathcal{R}_i$ );
        Tổng hợp kết quả;
    }
}
```

# III. Bài toán áp dụng

## 1. Tìm kiếm nhị phân

The Manhattan phone book has 1,000,000+ entries.

How is it possible to locate a name by examining just a tiny, tiny fraction of those entries?

wide at SuperPages.com

195 Car C

17 566-1282	Cartage New England Inc 26 Allen Ln Ipswich 01938.....	978 356-9960	Carter F 24 Hillock Ros 02131.....	617 327-1105	Carter Nella E 333 Maschots Av Bos 02115.....	617 267-6483
81 447-4101	Cartagema Lydia 18 Jewett Ros 02131.....	617 323-7639	Faye & Ricky 357 Columbus Av Bos 02116.....	617 437-7331	Nicholas S F 115 Randolph Av Mil 02186.....	617 698-5307
800 257-9981	Cartagema Avith 9 Bancroft Ros 02119.....	617 442-9780	Francis S 134 Temple W Ros 02132..	617 323-6781	Nick 21 Fairfield Bos 02116.....	617 267-5222
17 566-1282	B Hyd 02136.....	617 361-5253	Franklin & Anne 221 Mt Auburn Cam 02138.....	617 354-0798	Nick & Debbi 196 Herrick Rd Newton 02459.....	617 527-0480
17 364-5188	Jessica 50 Decatur Cha 02129.....	617 241-0152	Fred 42 Haverford Jam 02130.....	617 524-3078	Nicole 196 Herrick Rd Newton 02459.....	617 698-0713
361-0380	Lucilla 174 Harvard Cam 02139.....	617 491-5621	Fred 96 Hockley Rd Mil 02186.....	617 698-1343	Norman G 38 Chickatawbut Dor 02122.....	617 822-1203
17 566-4548	Melvin 501 Green Cam 02139.....	617 576-1061	G & R 8 Verdun Dor 02124.....	617 436-8906	P 94 Crestwood Pk Ros 02121.....	617 427-4754
17 628-8248	Carte Nicholas 18 Appleton Boston 02116.....	617 695-6996	G T 27 Franklin Av Som 02145.....	617 623-7121	P E 501 E Sixth S Bos 02127.....	617 268-4213
17 445-5116	Cartegena O 4 Millard Bos 02118.....	617 338-8219	Gayle 25 Frontenac Dor 02124.....	617 825-0322	P L 44 Hutchings Ros 02121.....	617 427-9170
17 822-2982	Carten Thos J Sr & Claire 1 Paradise Rd Mil 02186.....	617 698-6163	George 125 Nashua Bos 02114.....	617 367-9548	P R 91 Byrner Jam 02130.....	617 983-8692
17 427-5712	Thomas & Kathleen 50 Thompson Ln Mil 02186.....	617 696-6919	Carter Halliday Associate 107 S Street Bos 02111.....	617 456-1689	Paul & Constance 114 Anawan Av W Ros 02132.....	617 325-2036
17 569-2698	Carter A Ros 02131.....	617 327-2257	Carter Harry F 26 Running Bk Rd W Ros 02132.....	617 325-5465	Paul E 501 E Sixth St S Bos 02127.....	617 268-4546
17 667-5190	A Rosbury.....	617 442-5230	Carter Hide Co Inc 146 Summer Bos 02110.....	617 542-7987	Paul M 27 Union Bri 02135.....	617 787-2115
17 569-1417	A 31 Bethune Wy Roxbury 02119.....	617 442-1219	Carter Hilary 61 Harvey Cam 02140.....	617 876-2750	Carter Pile Driving Inc 17 Beaver Ct Frammingham 01702.....	781 235-8488
17 338-9110	A 260 Putnam Av Cambridge 02139.....	617 492-4174	Horace 241 Walnut Av Roxbury 02119.....	617 442-5307	Carter Prudence 46 Franklin Watertown 02172.....	617 393-3782
17 825-9195	A M 255 Maschots Av Bos 02115.....	617 266-7153	Howard Jr 26 Notre Dme Ros 02119.....	617 445-5552	Prudence 46 Franklin Watertown 02172.....	617 926-7063
17 296-1593	Adams 361 Centre St Mil 02186.....	617 698-9074	J Carl.....	617 354-2688	Reginald 106 Brunswick Dorchester 02121.....	617 541-2843
17 670-2078	Alice 108 Kilmarock Bos 02215.....	617 425-0193	J 15 Chatham Bro 02446.....	617 232-7990	Renee & Andrew 10 Walnut Bos 02108.....	617 720-3765
17 623-9001	Alice 45 Market Cambridge 02139.....	617 945-2711	J 518 Harvard Bro 02446.....	617 730-9483	Carter Rice Dowd Bulley Dutton Publishing 163 Main Wilmington 01887.....	800 638-1671
17 296-4725	Andrew F 62 Vinal Av Som 02143.....	617 625-7623	J 775 Vw Plow West Roxbury 02132.....	617 323-5574	Toll Free-Dial '1' & Then.....	800 638-1671
17 542-1521	Carter Anne MD 1101 Beacon Bro 02446.....	617 739-1022	Carter J Jacques MD 1 Brookline Pl Bro 02446.....	617 735-8787	Cust Svc-Industrial Prod 613 Main Wilmington Toll Free-Dial '1' & Then.....	800 619-7447
17 364-5232	Carter Athens 272 Newbury Boston 02116.....	617 536-6329	Carter J M 1410 Columbia Rd S Bos 02127.....	617 464-1040	Cust Svc-Printing 613 Main Wilmington Toll Free-Dial '1' & Then.....	800 648-7447
17 541-5649	B E 48 Gladeside Av Mat 02126.....	617 296-6911	Carter J M Ornamental Ironworks Call.....	617 436-5353	Headquarters 613 Main Wilmington 01887.....	978 988-7447
17 739-2662	Carter Barbara L MD Tufts-New England Medical Center Bos 02111.....	617 636-0051	Carter J Veal Co 48 Newmarket Sq Ros 02118.....	617 442-1775	Call.....	978 988-7447
17 879-0030	Carter Becky Bos 02114.....	617 523-4368	Carter James 1573 Cambridge St Cam 02138.....	617 492-1214	Ingalls Cronin 163 Main Wilmington 01887.....	800 638-1673
17 541-3948	Bernard J 112 Gladstone E Bos 02128.....	617 567-3430	James 182 Fisher Av Roxbury 02120.....	617 739-2193	Toll Free-Dial '1' & Then.....	800 638-1673
17 436-1513	Bithiah 25 Medway Dor 02124.....	617 298-8713	James 37 Gold Star Rd Cambridge 02140.....	617 876-8841	Carter Richard 1079 Commwth Av Brighton 02215.....	617 987-0836
17 569-4119	Blake 26 Mt Vernon Bos 02108.....	617 367-9931	Jas L 14 Roseberry Rd Mat 02126.....	617 361-0773	Richard A 97 Mt Vernon Bos 02108.....	617 566-7293
17 739-2662	Carter Broadcasting Co 20 Park Pkz Bos 02116.....	617 423-0210	Jane 114 Adena Rd Newton 02465.....	617 964-0435	Carter Richard A MD 170 Commwth Av Bos 02116.....	617 267-0710
17 879-0030	Carter & Burgess Consultants Inc 23 East St Cam 02141.....	617 225-0200	Jeffrey 41 Warren Av Bos 02116.....	617 426-5994	Carter Richard K 15 Mercer S Bos 02127.....	617 268-0448
17 541-3948	Carter C 2000 Commwth Av Bri 02135.....	617 782-2118	John 11 Mansfield Bri 02134.....	617 987-2163	Robert L 175 Ridchale Av Cam 02140.....	617 864-1535
17 436-1513	C 228 Faywood Av East Boston 02128.....	617 569-1545	John 327 Summer Bos 02110.....	617 423-4334	Roger 150 St Botolph Bos 02115.....	617 424-6148
17 569-4119	C 359 Harvard Cam 02138.....	617 491-4822	John 40 Westwind Rd Dor 02125.....	617 282-1235	Roy 44 Concord Av Cam 02138.....	617 491-6115
17 569-4119	C 610 Walk Hill Mat 02126.....	617 296-6392	June O 329 A Summit Av Bri 02135.....	617 734-6109	Royce 18 Seminary Cha 02129.....	617 241-0418
17 569-4119	C 610 Walk Hill Mat 02126.....	617 296-6392	K 38 Browning Av Dorchester 02124.....	617 265-8456		
17 569-4119	C 610 Walk Hill Mat 02126.....	617 296-6392	K 17 Esmond Dorchester 02121.....	617 282-1593		

# III. Bài toán áp dụng

**1. Tìm kiếm nhị phân**      To find the page containing Pat Reed's number...

while (Phone book is longer than 1 page)

Open to the middle page.

if “Reed” comes before the first entry,

Rip and throw away the 2<sup>nd</sup> half.

else

Rip and throw away the 1<sup>st</sup> half.

end

end

**Key idea of “phone  
book search”:  
repeated halving**



# III. Bài toán áp dụng

## 1. Tìm kiếm nhị phân

**What happens to the  
phone book length?**

Original:	3000	pages
After 1 rip:	1500	pages
After 2 rips:	750	pages
After 3 rips:	375	pages
After 4 rips:	188	pages
After 5 rips:	94	pages
:		
After 12 rips:	1	page

# III. Bài toán áp dụng

## 1. Tìm kiếm nhị phân

- Repeatedly halving the size of the “search space” is the main idea behind the method of **binary search**.
- An item in a sorted array of length **n** can be located with just  **$\log_2 n$**  comparisons.
- “Savings” is significant!

n	$\log_2(n)$
100	7
1000	10
10000	13

### III. Bài toán áp dụng

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98

Binary  
search:  
target  $x = 70$



L:

1

Mid:

6

R:




12

$v(\text{Mid}) \leq x$

So throw away the left  
half...

### III. Bài toán áp dụng

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



Binary  
search:  
target  $x = 70$

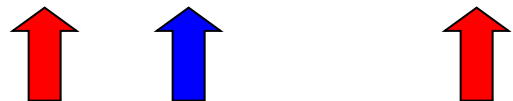
L: 6  
Mid: 9  
R: 12

$x < v(\text{Mid})$

So throw away the  
right half...

### III. Bài toán áp dụng

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



Binary  
search:  
target  $x = 70$

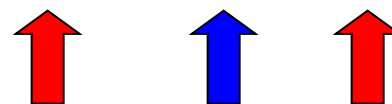
L: 6  
Mid: 7  
R: 9

$v(\text{Mid}) \leq x$

So throw away the left  
half...

### III. Bài toán áp dụng

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



Binary  
search:  
target  $x = 70$

L:

7

Mid:

8

R:

9

$v(\text{Mid}) \leq x$

So throw away the left  
half...

### III. Bài toán áp dụng

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



Binary  
search:  
target  $x = 70$

L: 8  
Mid: 8  
R: 9

Done because  
 $R - L = 1$

# III. Bài toán áp dụng

## 1. Tìm kiếm nhị phân

- Mô tả thuật toán:
  - Vào  $A[1..n]$
  - Ra: Chỉ số  $k = -1$  nếu không tìm thấy  
 $1 \leq k \leq n$  nếu tìm thấy
- Độ phức tạp thuật toán:  $O(\log_2 n)$

```
Tknp(a, x, Đầu, Cuối)  $\equiv$   
    If (Đầu > Cuối)  
        return 0 ; {dãy trống}  
    Else  
    {  
        Giữa = (Đầu + cuối) / 2;  
        If (x == a[Giữa])  
            return 1;  
        else  
            if (x > a[Giữa])  
                Tknp(a, x, Giữa + 1, Cuối) ;  
            else  
                Tknp(a, x, Đầu, Giữa - 1) ;  
    }  
}
```



# III. Bài toán áp dụng

## 1. Tìm kiếm nhị phân

- Cài đặt:

```
int tknp(int a[max],int x,int l, int r)
{
    int mid;
    if ( l > r )
        return 0;
    mid = (l+r)/2;
    if ( x == a[mid] )
        return 1;
    else
        if ( x > a[mid] )
            return tknp(a,x,mid+1,r);
        else
            return tknp(a,x,l,mid-1);
}
```

# III. Bài toán áp dụng

## 2. Tìm giá trị MIN, MAX

- Phát biểu bài toán: Cho mảng A có n phần tử. Tìm giá trị lớn nhất (MAX) và giá trị nhỏ nhất (MIN) trên mảng A.
- Tìm kiếm “nhị phân”:
  - Chia đôi mảng A, tìm kiếm MIN, MAX trên mỗi nửa sau đó tổng hợp kết quả trên hai nửa đó để tìm MIN, MAX của cả mảng A.
  - Nếu đoạn chia chỉ có một phần tử thì  $\text{MIN}=\text{MAX}=\text{phần tử đó}$ .

# III. Bài toán áp dụng

## 2. Tìm giá trị MIN, MAX

- Mô tả thuật toán:
  - Vào:  $A[l..r]$
  - Ra:  $MIN = \text{Min}(A[1], \dots, A[r])$   
 $MAX = \text{Max}(A[1], \dots, A[r])$

```
MinMax(a,l, r, Min, Max)
{
    if (l == r)
    {
        Min = a[l];
        Max = a[l];
    }
    Else
    {
        MinMax(a,l, (l+r) / 2, Min1, Max1);
        MinMax(a,(l+r) / 2 + 1, r , Min2, Max2);
        If (Min1 < Min2)
            Min = Min1;
        Else
            Min = Min2;
        If (Max1 > Max2)
            Max = Max1
        Else
            Max = Max2;
    }
}
```

# III. Bài toán áp dụng

## 2. Tìm giá trị MIN, MAX

- Độ phức tạp thuật toán:

Gọi  $T(n)$  là số phép toán so sánh

$$T(n) = \begin{cases} T(n/2) + T(n/2) + 2 & ; n > 2 \\ 1 & ; n = 2 \\ 0 & ; n = 1 \end{cases}$$

Với  $n = 2^k$ , thì :

$$\begin{aligned} T(n) &= 2 + 2T(n/2) = 2 + 2^2 + 2^2 T(n/2^2) = \dots = 2^{k-1} T(2) + \sum_{i=1}^{k-1} 2^i \\ &= \sum_{i=1}^k 2^i - 2^{k-1} = 2^{k+1} - 2^{k-1} - 2 = \frac{3n}{2} - 2. \end{aligned}$$

Vậy  $T(n) \in O(n)$ .

# III. Bài toán áp dụng

## 2. Tìm giá trị MIN, MAX

- Cài đặt:

```
void MinMax(int a[.], int l, int r, int &Min, int &Max )
{   int Min1,Min2,Max1,Max2;
    if (l == r )
    {   Min = a[l];
        Max= a[l];   }
    else
    {   MinMax(a,l,(l+r)/2 , Min1, Max1);
        MinMax(a,(l+r)/2 + 1,r, Min2, Max2);
        if (Min1 < Min2)
            Min = Min1;
        else
            Min = Min2;
        if (Max1 > Max2)
            Max = Max1;
        else
            Max = Max2;
    }
}
```

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- Phát biểu bài toán: Cho mảng gồm  $n$  phần tử  $A[1..n]$ , sắp xếp mảng  $A$  theo thứ tự tăng dần
- **Ý tưởng:**
  - Nếu có hai dãy  $a$  và  $b$  đã được sắp xếp, tiến hành trộn hai dãy này thành dãy  $c$  đã được sắp xếp.
  - Nếu chia nhỏ mảng cần sắp xếp thành các đoạn 1 phần tử thì nó là đoạn được sắp xếp
  - Tiến hành ghép các đoạn nhỏ thành các đoạn lớn đã được sắp xếp

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

If I have two helpers, I'd...

- Give each helper half the array to sort
- Then I get back the sorted subarrays and *merge* them.

What if those two helpers each had two sub-helpers?

And the sub-helpers each had two sub-sub-helpers?  
And...

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

H	E	M	G	B	K	A	Q	F	L	P	D	R	C	J	N
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

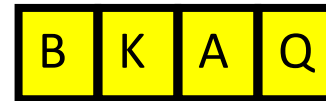
H	E	M	G	B	K	A	Q
---	---	---	---	---	---	---	---

F	L	P	D	R	C	J	N
---	---	---	---	---	---	---	---



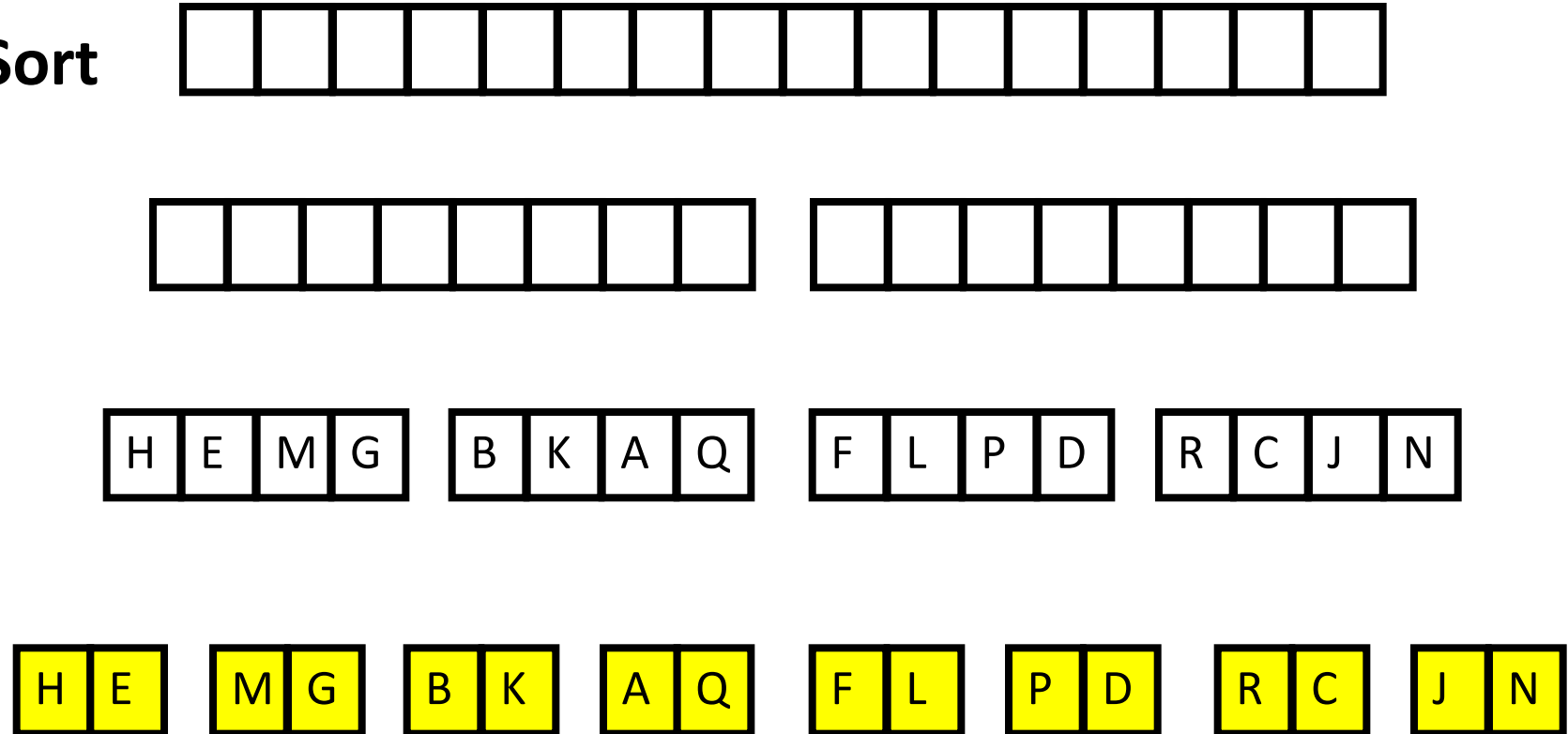
# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



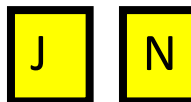
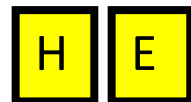
# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



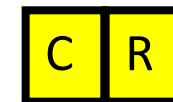
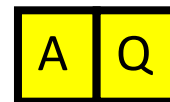
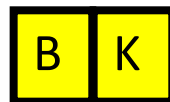
# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



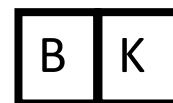
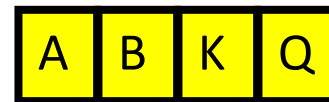
# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



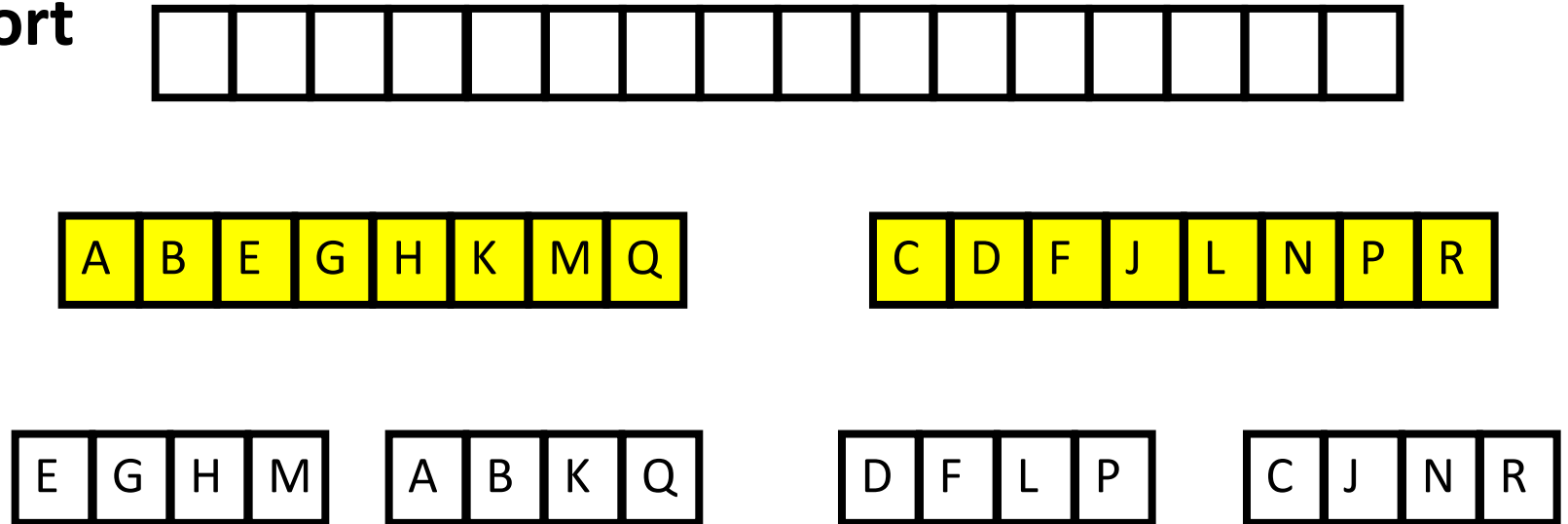
# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



# III. Bài toán áp dụng

## 3. Thuật toán MergeSort



# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

A	B	E	G	H	K	M	Q
---	---	---	---	---	---	---	---

C	D	F	J	L	N	P	R
---	---	---	---	---	---	---	---

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

H	E	M	G	B	K	A	Q	F	L	P	D	R	C	J	N
A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R



# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- **Ý tưởng thao tác trộn:**

- Duyệt trên dãy a tại vị trí i
- Duyệt trên dãy b tại vị trí j
- Nếu  $a[i] > b[j]$  thì thêm  $b[j]$  vào trong dãy c tăng biến j ngược lại thêm  $a[i]$  vào dãy và tăng biến i
- Nếu một trong hai dãy hết trước tiến hành đưa toàn bộ dãy còn lại vào trong dãy c
- Áp dụng trong trường hợp a, b là hai đoạn của mảng
  - $a[l..t]$ ,  $a[t+1..r]$
  - $c[l..r]$
- Để thuận tiện trong xử lý tiến hành chuyển mảng đã sắp xếp về mảng a

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- Input:  $a[l..t]$ ,  $a[t+1..r]$  đã được sắp xếp
- Output:  $a[l..r]$  được sắp xếp không giảm

1.  $i=l$

2.  $j=t+1$

3.  $p=l$ ;

4. while ( $i \leq t \ \&\& \ j \leq r$ )

    a. if( $a[i] < a[j]$ )

$c[p]=a[i]$

$i++$

    b. Else

$c[p]=a[j]$ ;

$j++$

    c.  $p++$

5. while ( $i \leq t$ )

$c[p]=a[i]$

$i++$

$p++$

6. while ( $j \leq r$ )

$c[p]=a[j]$

$j++$

$p++$

7. for ( $i=l$ ;  $i \leq r$  ; $i++$ )

$a[i]=c[i]$ ;

# III. Bài toán áp dụng

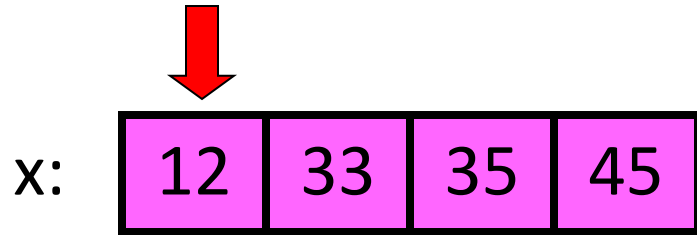
## 3. Thuật toán MergeSort


12	33	35	45
----	----	----	----

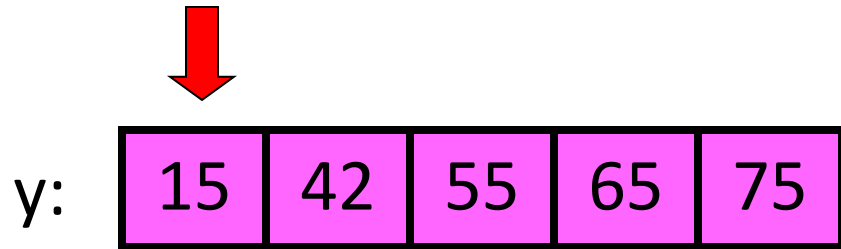
15	42	55	65	75
----	----	----	----	----


12	15	33	35	42	45	55	65	75
----	----	----	----	----	----	----	----	----

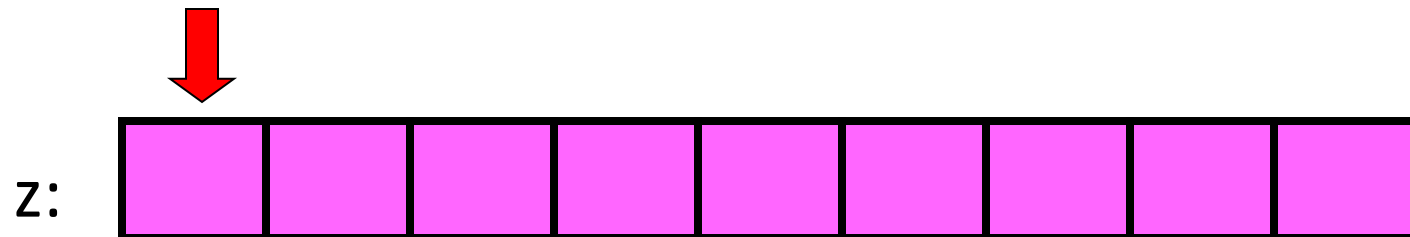
# Merge




ix: 



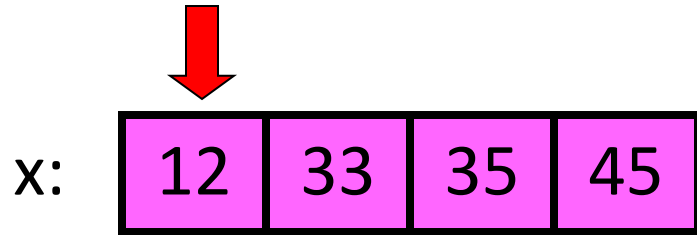
iy: 




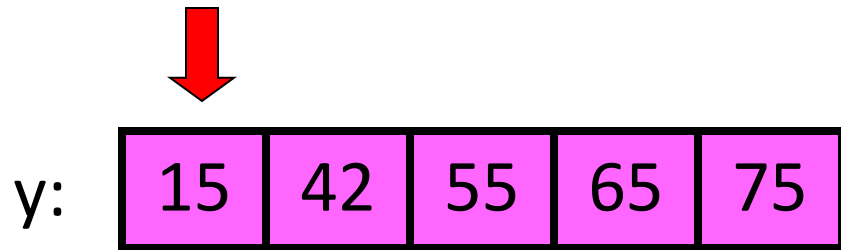
iz: 


$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???

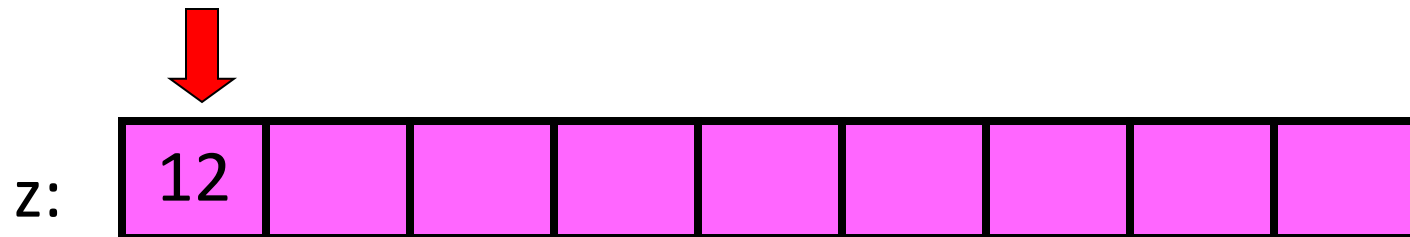
# Merge




ix: 



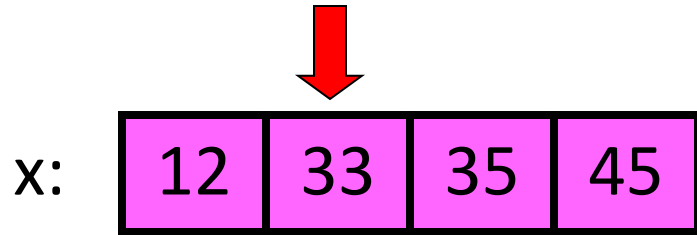
iy: 




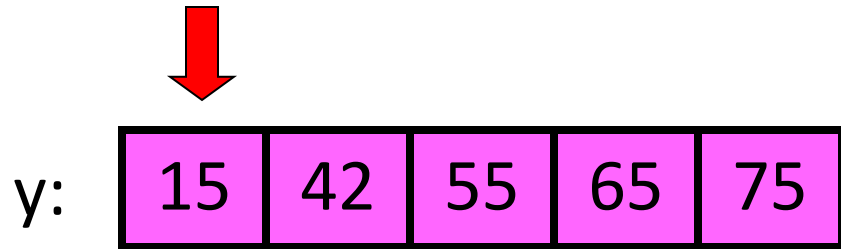
iz: 


$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  YES

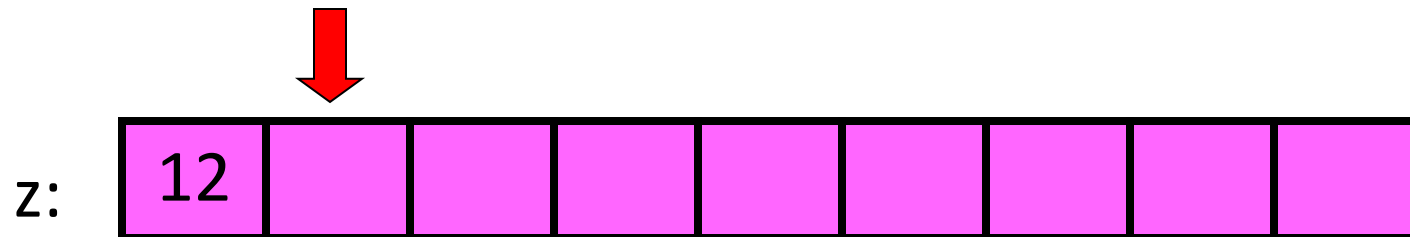
# Merge




ix: 



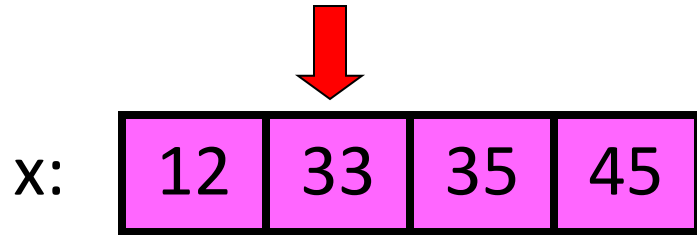
iy: 




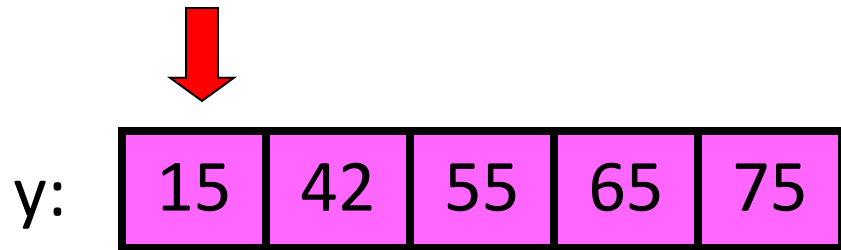
iz: 


$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???

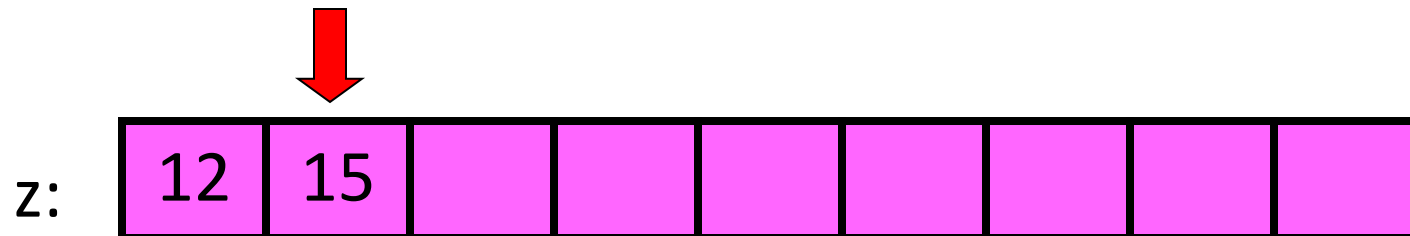
# Merge




ix: 



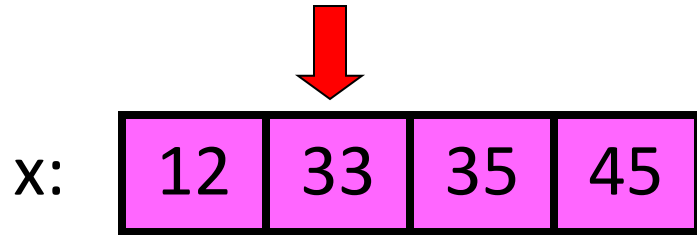
iy: 




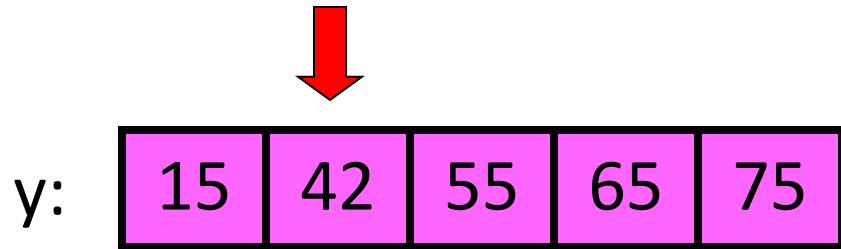
iz: 


$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  **NO**

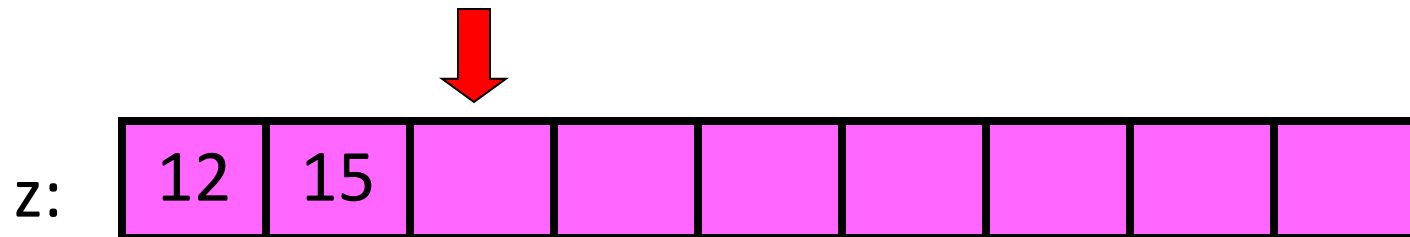
# Merge




ix: 



iy: 

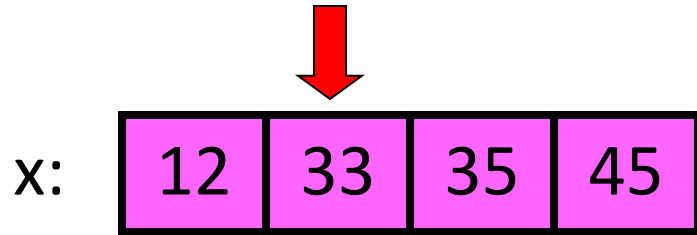



iz: 

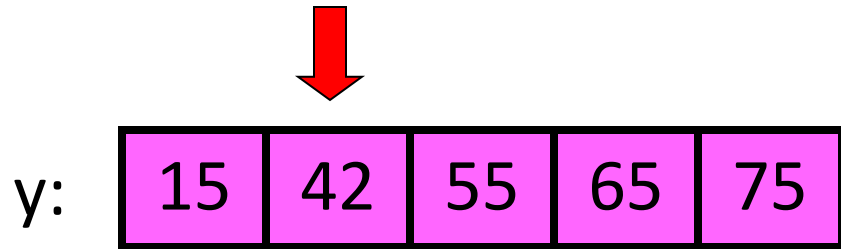
$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???




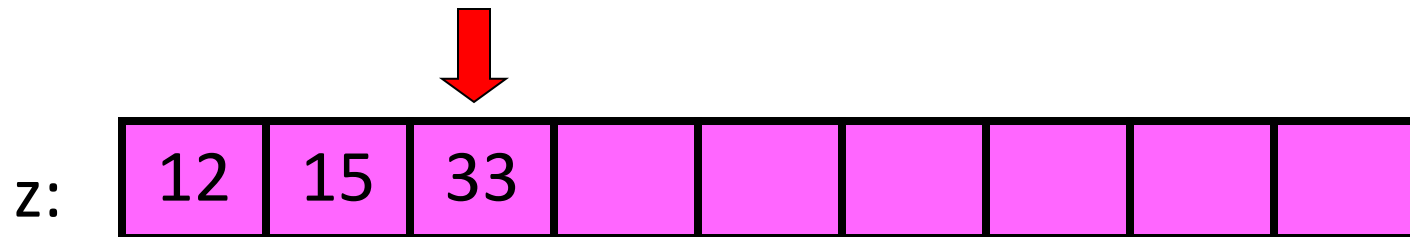
# Merge




ix: 



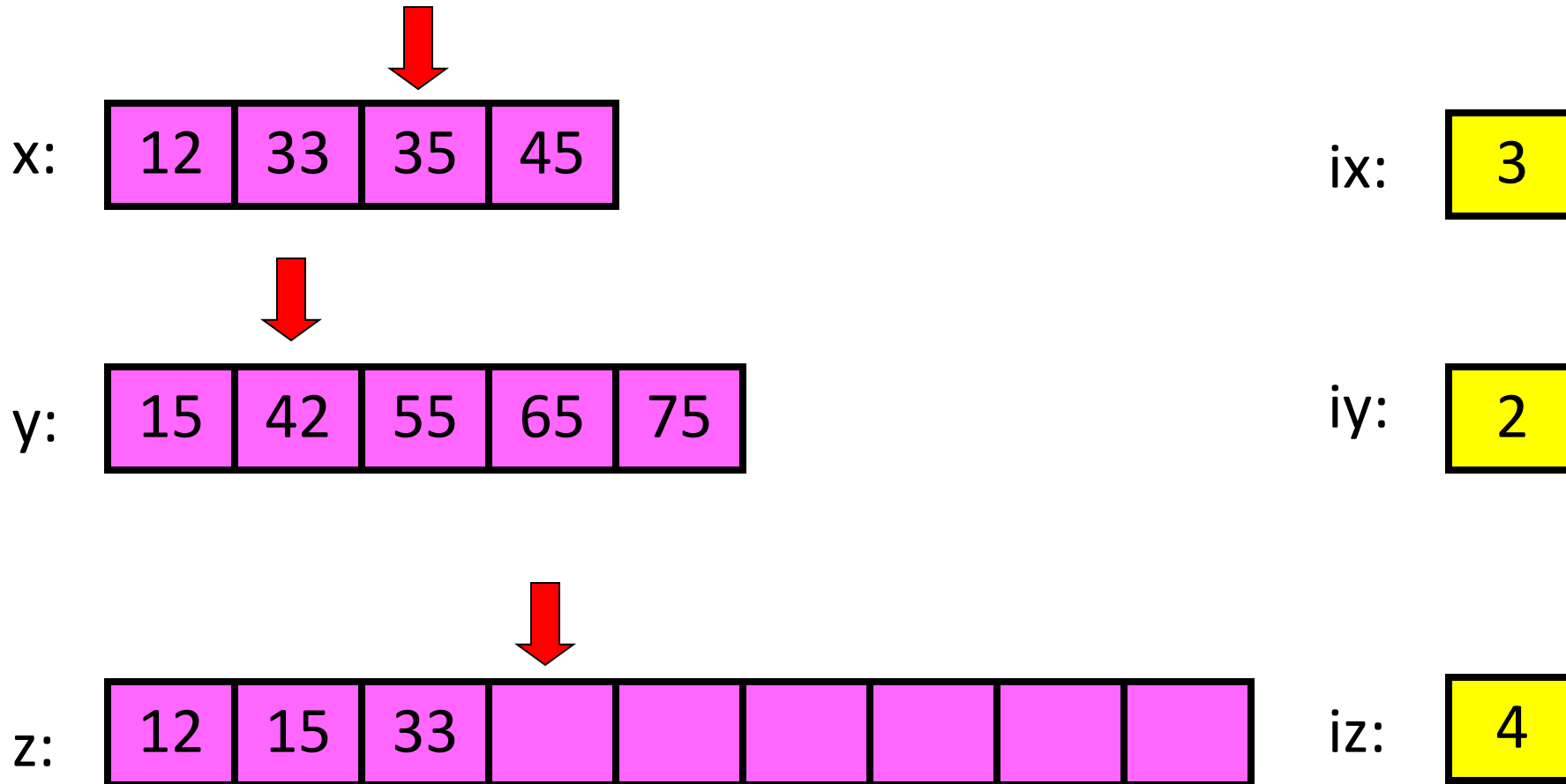
iy: 



iz: 

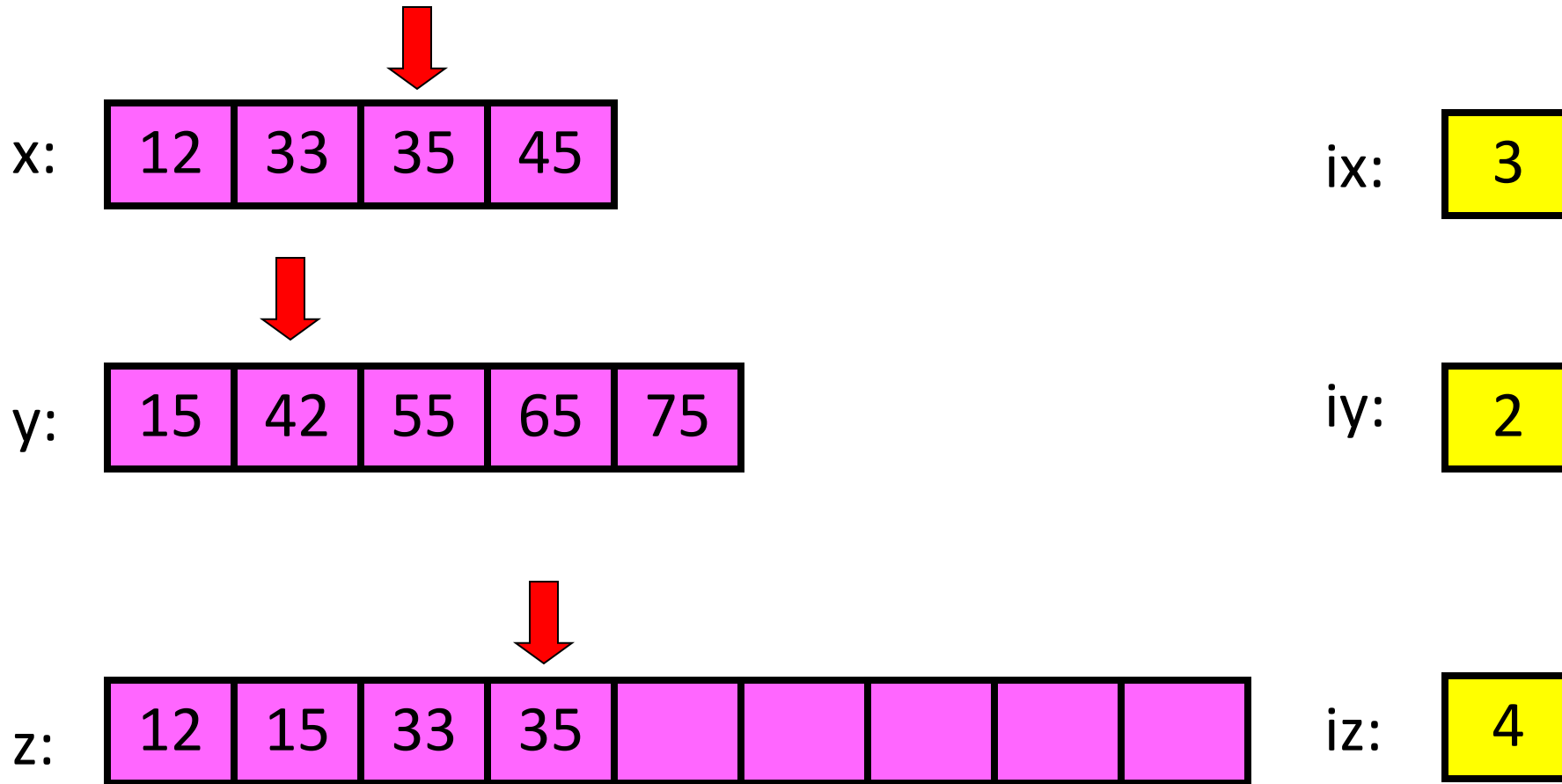
$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  YES

# Merge



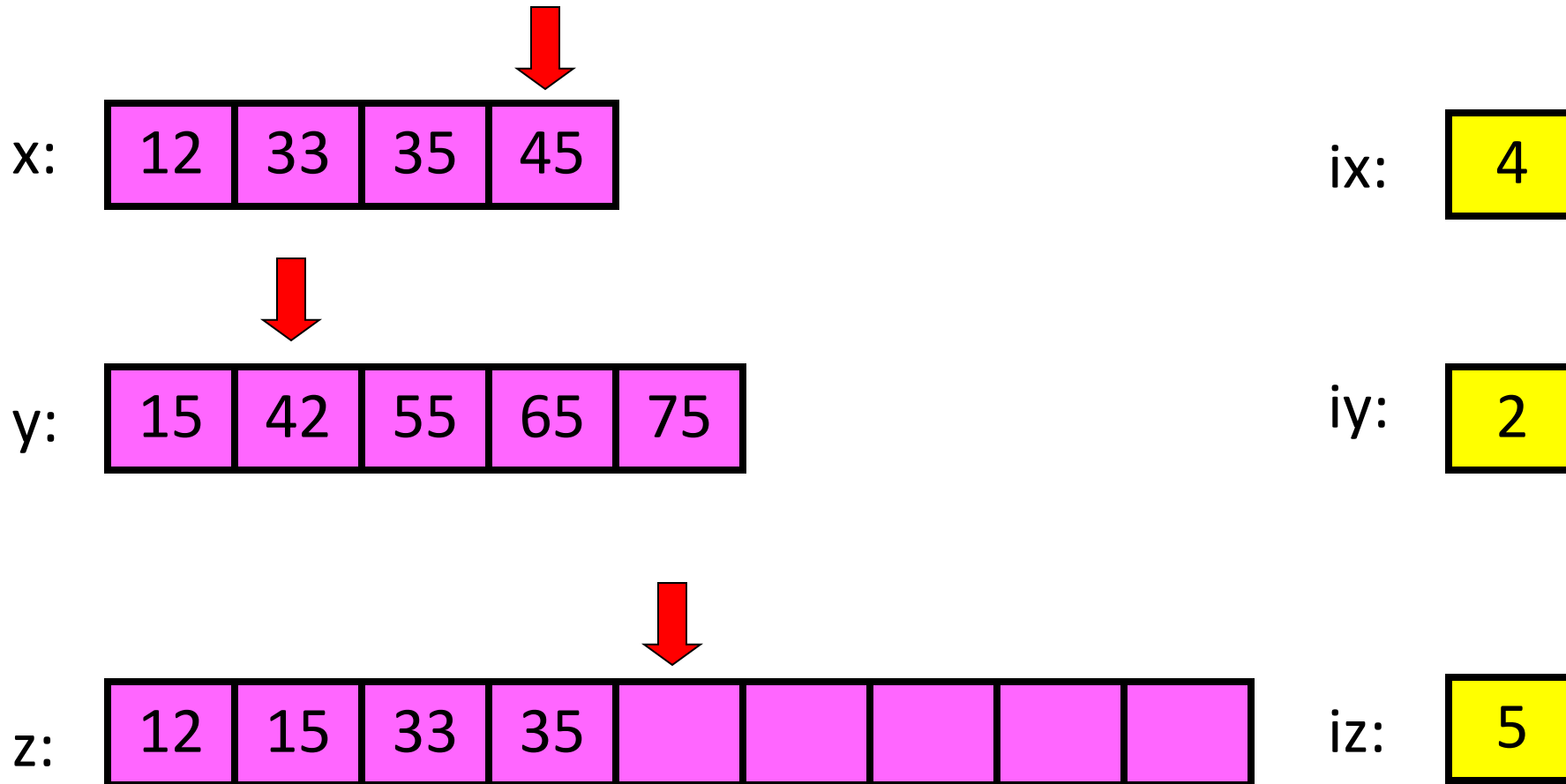
$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???

# Merge



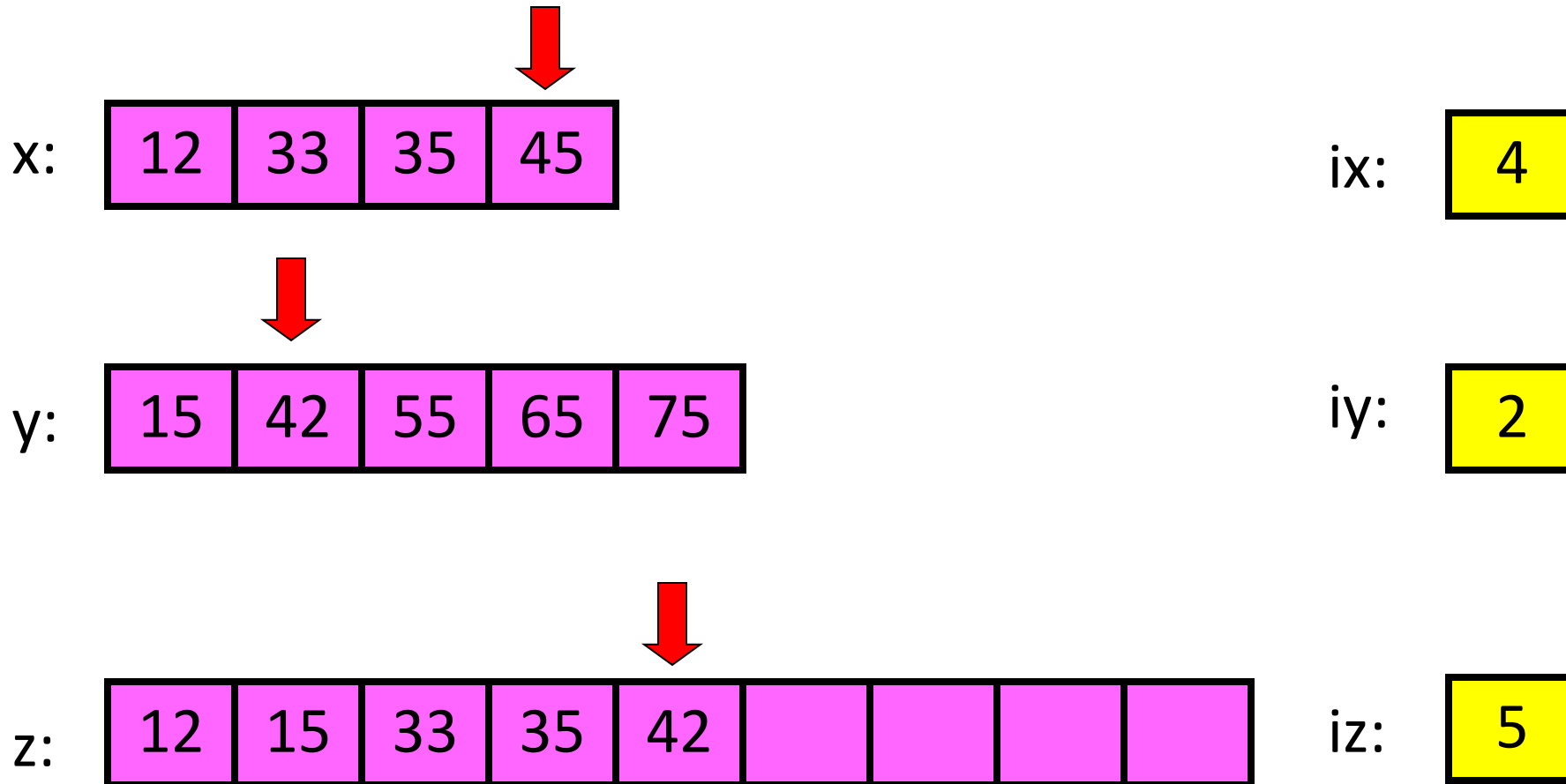
$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  YES

## Merge



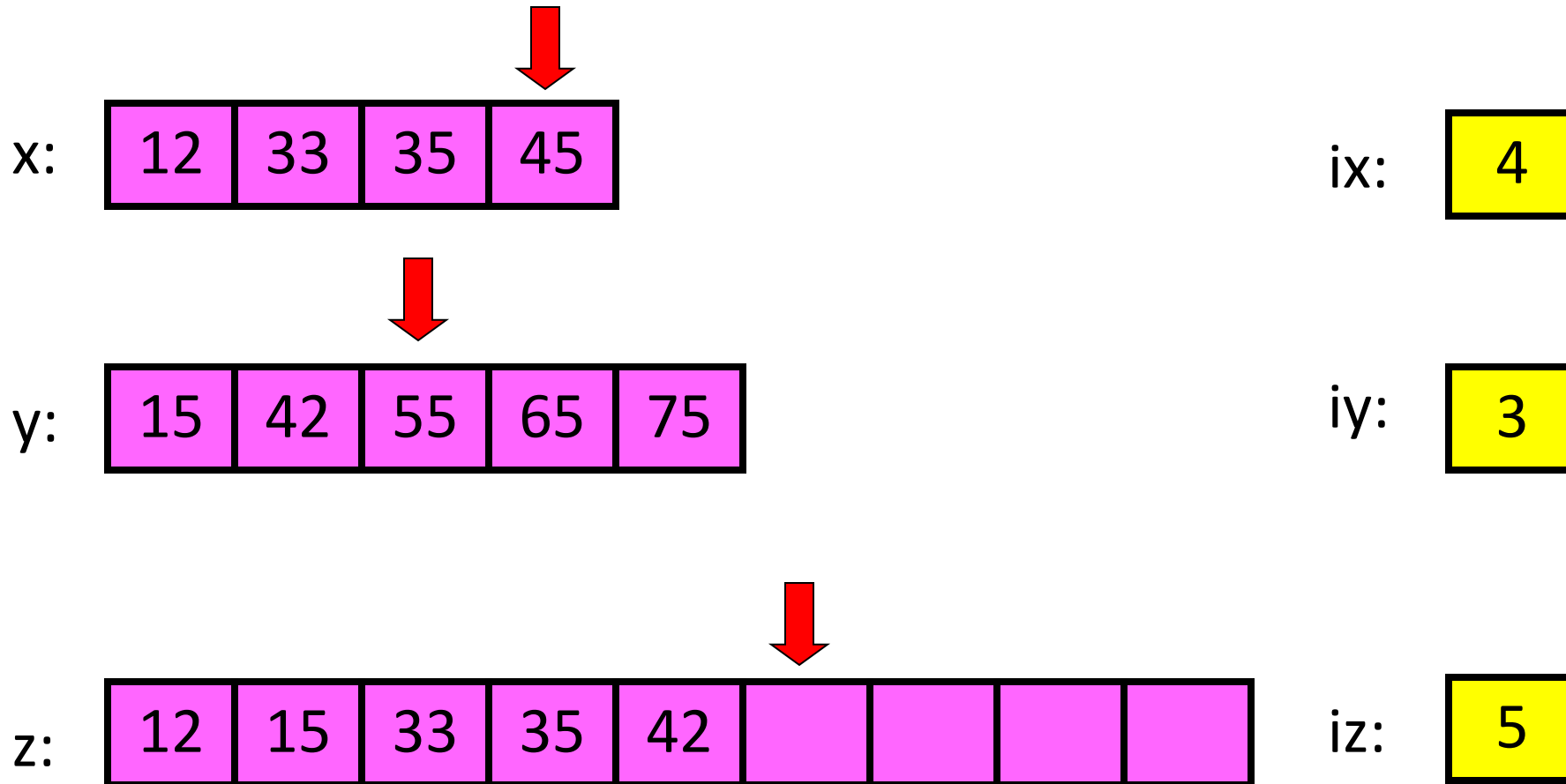
$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???

## Merge



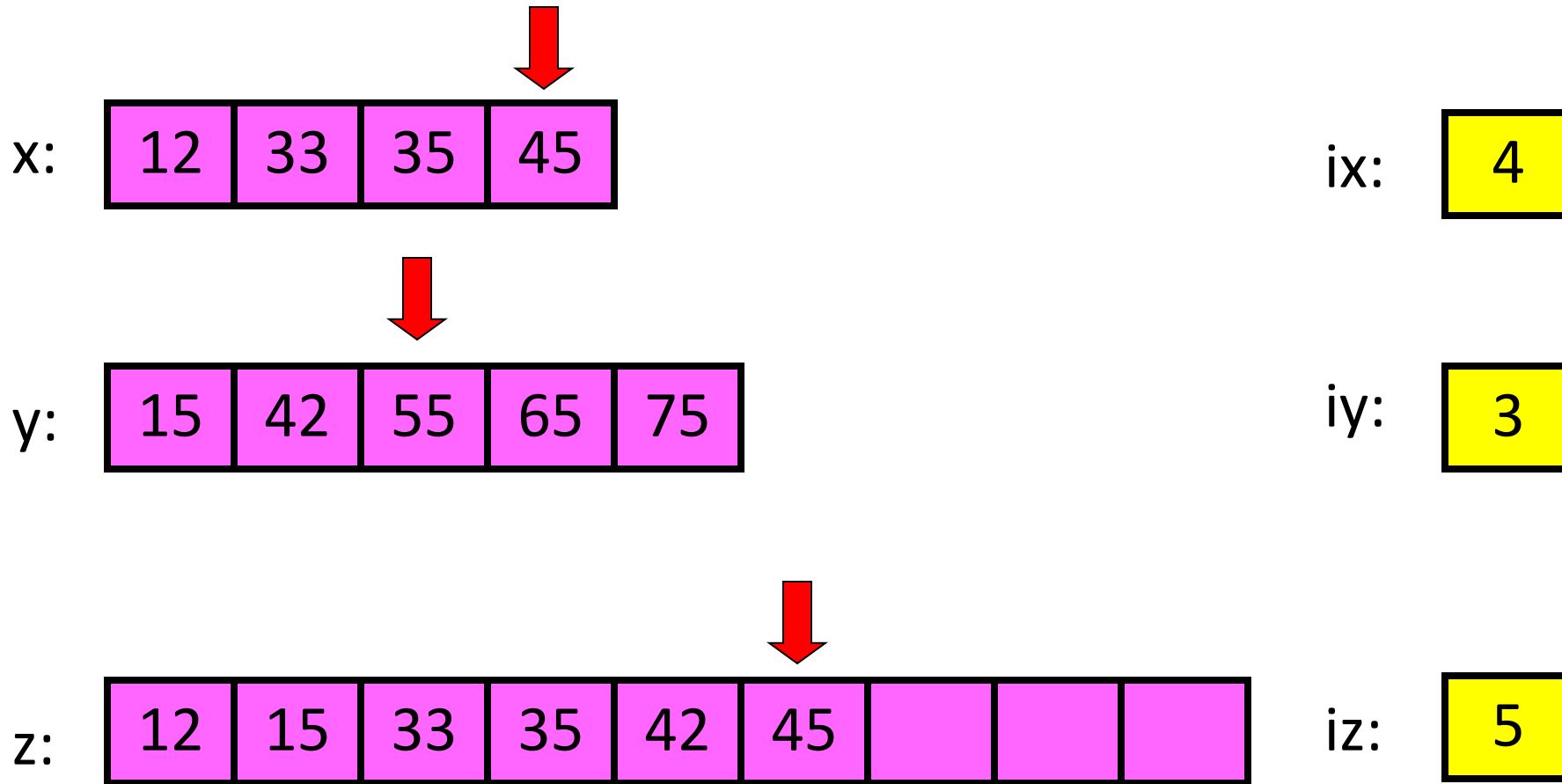
$ix \leq 4$  and  $iy \leq 5$ :  $x(ix) \leq y(iy)$  **NO**

## Merge



$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  ???

## Merge



$ix \leq 4$  and  $iy \leq 5$ :  $x[ix] \leq y[iy]$  YES

# Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

3
---



z: 

12	15	33	35	42	45			
----	----	----	----	----	----	--	--	--

iz: 

6
---

$ix > 4$



## Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

3
---



z: 

12	15	33	35	42	45	55		
----	----	----	----	----	----	----	--	--

iz: 

6
---

$ix > 4$ : take  $y(iy)$

# Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

4
---



z: 

12	15	33	35	42	45	55		
----	----	----	----	----	----	----	--	--

iz: 

8
---

$iy \leq 5$

# Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

4
---



z: 

12	15	33	35	42	45	55	65	
----	----	----	----	----	----	----	----	--

iz: 

8
---

$iy \leq 5$

## Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

5
---



z: 

12	15	33	35	42	45	55	65	
----	----	----	----	----	----	----	----	--

iz: 

9
---

$iy \leq 5$

# Merge



x: 

12	33	35	45
----	----	----	----

ix: 

5
---



y: 

15	42	55	65	75
----	----	----	----	----

iy: 

5
---



z: 

12	15	33	35	42	45	55	65	75
----	----	----	----	----	----	----	----	----

iz: 

9
---

$iy \leq 5$

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- Thuật toán sắp xếp trộn mergesort
- Input:  $a[l..r]$
- Output:  $a[l..r]$  đã được sắp xếp
  1. if( $l \geq r$ ) return ;
  2.  $t = (l+r)/2$
  3. mergesort( $l, t$ );
  4. mergesort( $t+1, r$ );
  5. merge( $a[l..t], a[t+1..r]$ );

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- Thuật toán sắp xếp trộn mergesort
- Input:  $a[l..r]$
- Output:  $a[l..r]$  đã được sắp xếp
  1. if( $l \geq r$ ) return ;
  2.  $t = (l+r)/2$
  3. mergesort( $l, t$ );
  4. mergesort( $t+1, r$ );
  5. merge( $a[l..t], a[t+1..r]$ );

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

1	3	7	8	2	6	9
1	3	7	8	2	6	9
1	2	3	6	7	8	9

# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

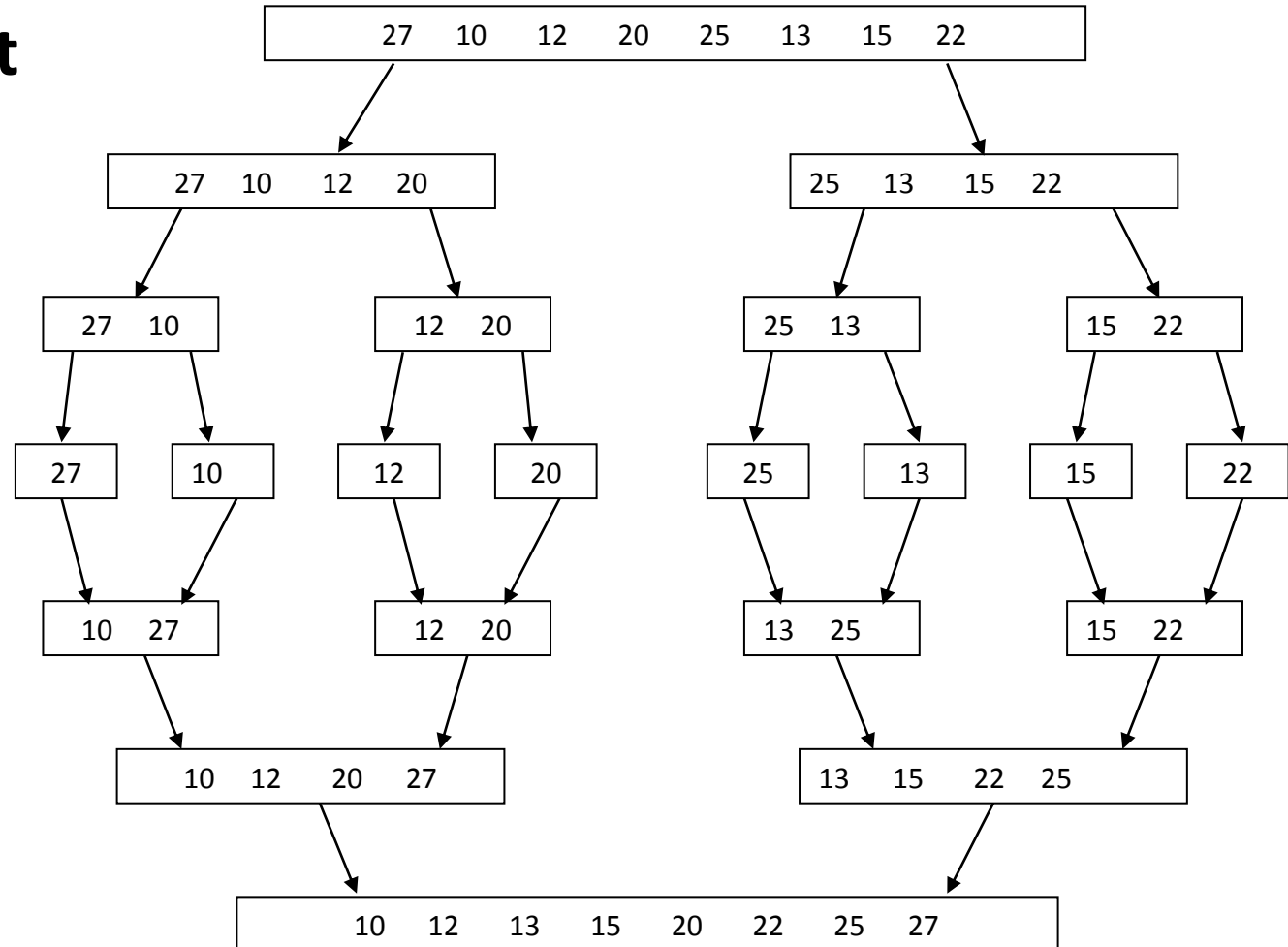
- Đánh giá độ phức tạp
- Số phép so sánh:  $n \cdot \log(n)$
- Số phép gáp:  $2 \cdot n \cdot \log(n)$
- Số phép gán chỉ số:  $2 \cdot n$
- Độ phức tạp phép toán:  $O(n \log(n))$



# III. Bài toán áp dụng

## 3. Thuật toán MergeSort

- Ví dụ



# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Phát biểu bài toán: Cho mảng gồm  $n$  phần tử  $A[1..n]$ , sắp xếp mảng  $A$  theo thứ tự tăng dần.
- Ý tưởng:
  - Cho một dãy, chọn một phần tử ở giữa, chia đoạn thành 2 phần
  - Chuyển các phần tử nhỏ, hoặc bằng đến trước, các phần tử lớn hơn về sau
  - Sẽ được nửa đầu bé hơn nửa sau
  - Lặp lại việc chuyển đổi cho các phần tử nửa đầu, và nửa sau đến lúc số phần tử là 1

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Phát biểu bài toán: Cho mảng gồm  $n$  phần tử  $A[1..n]$ , sắp xếp mảng  $A$  theo thứ tự tăng dần.
- Ý tưởng:
  - Thuật toán ban đầu là chia: cố gắng chia thành hai đoạn khác nhau
  - Trị: thực hiện các thuật toán sắp xếp trên các đoạn con
  - Thực hiện kết hợp: thuật toán tự kết hợp kết quả

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Phân đoạn (chia):
  - Chọn một phần tử chốt  $x$  (đầu tiên)
  - Duyệt từ vị trí tiếp theo sang phải tìm vị trí phần tử đầu tiên  $\geq x$ ,  $i$
  - Duyệt từ phải sang trái, tìm vị trí phần tử đầu tiên  $< x$ ,  $j$
  - Nếu  $i < j$  thì hoán đổi vị trí
  - Tiếp tục đến lúc  $j < i$

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Thuật toán: partition
- Input:  $A[l..r]$ ,  $l, r$ : đoạn cần phân chia
- Output:  $A[l..r]$ ,  $i$  chỉ số phân chia
  1.  $X = a[l]$
  2.  $i = l + 1$ ;
  3.  $j = r$ ;
  4. While ( $i < j$ )
    - a. While ( $i < j \ \&\& \ a[i] < x$ )  $i++$
    - b. While ( $j \geq i \ \&\& \ a[j] \geq x$ )  $j--$
    - c. If ( $i < j$ ) swap( $a[i], a[j]$ )
  5. Swap( $a[l], a[j]$ )
  6. Return  $j$ ;

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Thuật toán: partition
- Input:  $A[l..r]$ ,  $l, r$ : đoạn cần phân chia
- Output:  $A[l..r]$ ,  $i$  chỉ số phân chia

1.  $X = a[l]$
2.  $i = l + 1$ ;
3.  $j = r$ ;
4. While ( $i < j$ )
  - a. While ( $i < j$  &&  $a[i] < x$ )  $i++$
  - b. While ( $j >= i$  &&  $a[j] >= x$ )  $j--$
  - c. If ( $i < j$ ) swap( $a[i], a[j]$ )
5. Swap( $a[l], a[j]$ )
6. Return  $j$ ;

i	j	0	1	2	3	4	5	6
2	4	3	1	7	8	2	6	9
3	2	3	1	2	8	7	6	9
KQ		2	1	3	8	7	6	9

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Thuật toán: quicksort
- Input:  $A[l..r]$ : đoạn cần sắp xếp
- Output:  $A[l..r]$  đã sắp xếp
  1. If( $l \geq r$ ) return;
  2.  $i = \text{partition}(A, l, r)$
  3.  $\text{quicksort}(A, l, i-1)$
  4.  $\text{quicksort}(A, i+1, r)$

# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Thuật toán: quicksort
- Input: A[l..r]: đoạn cần sắp xếp
- Output: A[l..r] đã sắp xếp
  1. If(l>=r) return;
  2. i=partition(A,l,r)
  3. quicksort(A,l,i-1)
  4. quicksort(A,i+1,r)

A	0	1	2	3	4	5	6
	3	1	7	8	2	6	9
Part	3	1	2	8	7	6	9
	2	1	3	8	7	6	9
Part	2	1		8	7	6	9
	1	2		6	7	8	9
Part	1			6	7		9
				6	7		
					7		
	1	2	3	6	7	8	9



# III. Bài toán áp dụng

## 4. Thuật toán QuickSort

- Đánh giá độ phức tạp
  - Số phép toán gán giá trị:  $3 * n/2 * h$
  - Số phép toán so sánh:  $n * h$
  - Số phép toán gán chỉ số:  $n * h$
- Trường hợp xấu nhất:  $h=n$
- Trường hợp trung bình:  $h = \log(n)$
- Độ phức tạp trường hợp xấu nhất:  $O(n^2)$
- Độ phức tạp trường hợp trung bình:  $O(n\log(n))$

# IV. Bài tập

Cho mảng  $A=\{3, 5, 8, 9, 4, 2, 7, 5, 3,9,8\}$

1. Thực hiện từng bước thuật toán MIN, MAX với mảng A.
2. Thực hiện thuật toán QuickSort và thể hiện kết quả từng bước với mảng A.
3. Thực hiện từng bước thuật toán tìm kiếm nhị phân các giá trị  $x=5, 6, 7$  với mảng đã sắp xếp ở bài 2.
4. Thực hiện thuật toán MergeSort và thể hiện kết quả từng bước với mảng A.
5. Cài đặt thuật toán tìm kiếm nhị phân, đánh giá bằng thực nghiệm và so sánh với lý thuyết.
6. Cài đặt thuật toán MIN-MAX, đánh giá bằng thực nghiệm và so sánh với lý thuyết.
7. Cài đặt chương trình QuickSort, đánh giá bằng thực nghiệm và so sánh với lý thuyết.
8. Cài đặt chương trình MergeSort, đánh giá bằng thực nghiệm và so sánh với lý thuyết.
9. Thử nghiệm QuickSort và MergeSort trên cùng các bộ dữ liệu, so sánh thời gian thực hiện các thuật toán đó.

# NỘI DUNG BÀI HỌC

- I. Giới thiệu
- II. Lược đồ chung
- III. Bài toán áp dụng
- IV. Bài tập