

### THUẬT TOÁN MERGE SORT

- 1. Hồ Thái Ngọc
- 2. ThS. Võ Duy Nguyên
- 3. TS. Nguyễn Tấn Trần Minh Khang



### BÀI TOÁN DẪN NHẬP 01



- Bài toán: Cho mảng một chiều các số nguyên a có n phần tử. Hãy tạo mảng b từ mảng a sao cho b chứa toàn số chẵn của mảng a.
- Ví dụ:















```
11. void TaoMang(int a[], int n, int b[], int& m)

12. {
13. m = 0;
14. for(int i=0; i<n; i++)
15. if(a[i]%2==0)
16. {
    Trả lời đúng: Mảng b hiện có m phần tử. Các phần tử trong mảng b được đánh chỉ số từ 19. }
    0 cho đến m-1.
```









```
11.void TaoMang(int a[], int n, int b[], int& m)
12.{
13.<sub>I</sub>
       \mathsf{m} = 0;
                                   Mảng b hiện có m phần tử.
14. l
       for(int i=0; i<n; i++)
                                   Các phần tử trong mảng b
            if(a[i]%2==0)
15.
                                   được đánh chỉ số từ 0 cho
16.
                                   đến m-1.
17.
                b[m] = a[i];
                                   Phần tử a[i] sẽ được thêm
18.
                m++;
                                   vào cuối mảng b.
19.
20.}
```



```
11.void TaoMang(int a[], int n, int b[], int& m)
12.{
13. I
       m = 0;
                                Mảng b hiện có m phần tử.
14.
       for(int i=0; i<n; i++)
                                Các phần tử trong mảng b
           if(a[i]%2==0)
15.
                                được đánh chỉ số từ 0 cho
16.
               b[m++] = a[i];
                                đến m-1.
17.}
                                Phần tử a[i] sẽ được thêm
                                vào cuối mảng b.
```



### BÀI TOÁN DẪN NHẬP 02



- Bài toán: Cho 2 mảng một chiều các số nguyên a và b được sắp thứ tự tăng dần. Hãy trộn mảng a và mảng b lại với nhau để được mảng c sắp thứ tự tăng dần.
- Ví du:

Mång a 15 27 32 64

Mång b -5 1 18 29 37 56 80 93

Mång c | -5 | 1 | 15 | 18 | 27 | 29 | 32 | 37 | 56 | 64 | 80 | 93

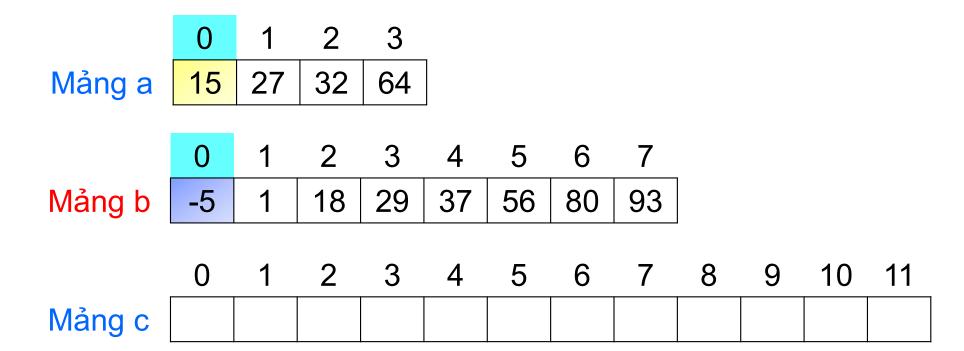




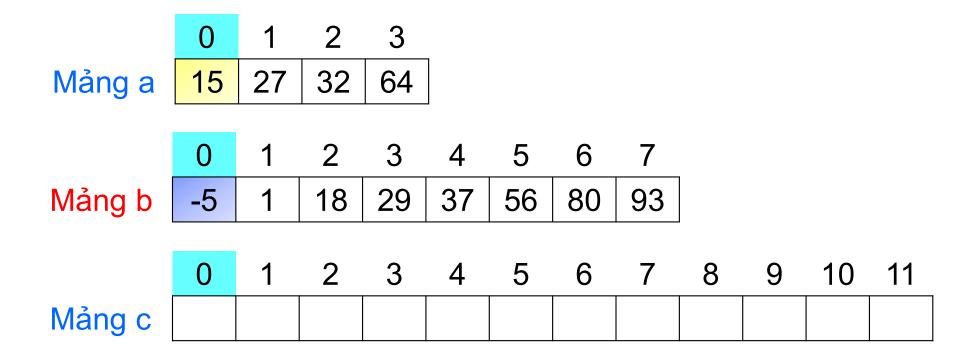


	0	1	2	3	_								
Mảng a	15	27	32	64									
		4			_	_		_					
	0	1	2	3	4	5	6						
Mảng b	-5	1	18	29	37	56	80	93					
	0	1	2	3	4	5	6	7	8	9	10	11	
Mảng c													











	0	1	2	3	_							
Mảng a	15	27	32	64								
					_							
	0	1	2	3	4	5	6	7				
Mảng b	-5	1	18	29	37	56	80	93				
	0	1	2	3	4	5	6	7	8	9	10	11
Mảng c	-5											



	0	1	2	3								
Mảng a	15	27	32	64								
					_							
	0	1	2	3	4	5	6	7				
Mảng b	-5	1	18	29	37	56	80	93				
				ı								
	0	1	2	3	4	5	6	7	8	9	10	11
Mảng c	-5	1										



	0	1	2	3	_								
Mảng a	15	27	32	64									
				•	_								
	0	1	2	3	4	5	6	7					
Mảng b	-5	1	18	29	37	56	80	93					
	0	1	2	3	4	5	6	7	8	9	10	11	
Mảng c	-5	1	15										



	0	1	2	3	_								
Mảng a	15	27	32	64									
	0	1	2	3	4	5	6	7					
Mảng b	-5	1	18	29	37	56	80	93					
'									l				
	0	1	2	3	4	5	6	7	8	9	10	11	
Mảng c	-5	1	15	18									









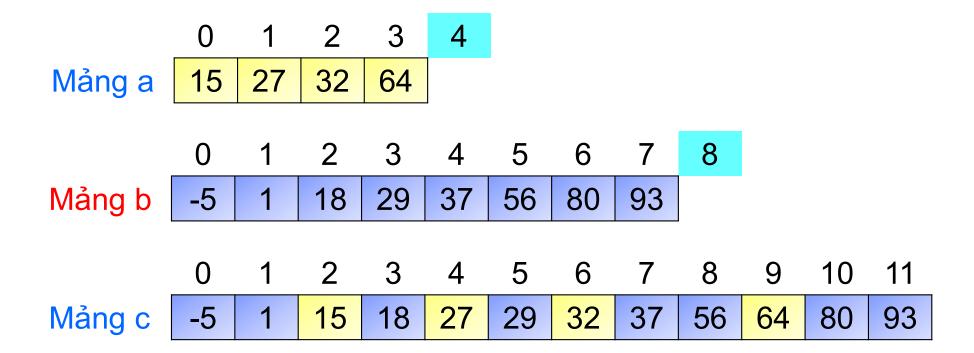


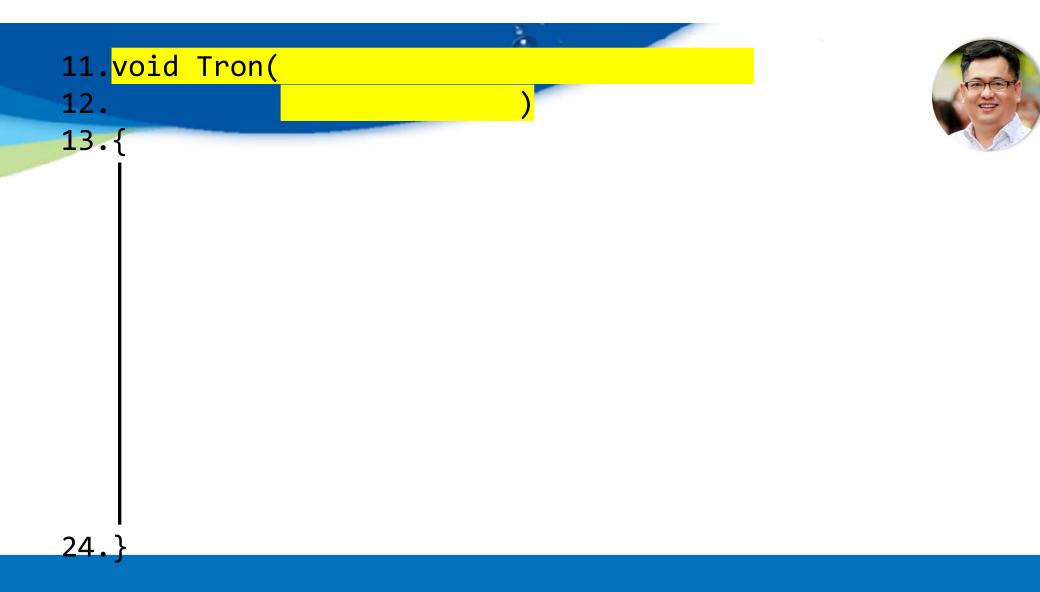












```
11.void Tron(int a[],int n,
12.
13.{
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
12.
13.{
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
            int c[],int p)
12.
13.{
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
       int j = 0;
15.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
       int j = 0;
15.
16.
      p = 0;
24.
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
       int j = 0;
15.
16.
     p = 0;
       while(
17.
18.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
       int j = 0;
15.
16.
     p = 0;
      while( i>=n
17.
18.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
            int c[],int &p)
12.
13.{
14.
       int i = 0;
       int j = 0;
15.
16.
    p = 0;
    while( i>=n j>=m )
17.
18.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
       while( i \ge n \&\& j \ge m)
17.
18.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n && j>=m))
18.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
       p = 0;
17.
       while(!(i>=n && j>=m))
18.
           if (
                               a[i]<b[j])
19.
20.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
    p = 0;
17.
       while(!(i>=n && j>=m))
18.
                            a[i]<b[j])
19.
           if (
20.
              c[p] = a[i];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n && j>=m))
18.
                             a[i]<b[j])
19.
           if (
20.
              c[p++] = a[i];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n && j>=m))
18.
                              a[i]<b[j])
19.
           if (
20.
               c[p++] = a[i++];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
       while(!(i>=n && j>=m))
17.
18.
                              a[i]<b[j])
19.
           if (
20.
               c[p++] = a[i++];
21.
           else
22.
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
                              a[i]<b[j])
19.
           if (
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p] = b[j];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
                              a[i]<b[j])
19.
           if (
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p++] = b[j];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
                              a[i]<b[j])
19.
           if (
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p++] = b[j++];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
              int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
19.
            if (i<n && j<m && a[i]<b[j])</pre>
20.
                c[p++] = a[i++];
21.
           else
22.
                c[p++] = b[j++];
23.
24.}
```



```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
19.
           if (i<n && j<m && a[i]<b[j]) || (j>=m)
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p++] = b[j++];
23.
24.}
```

```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
19.
           if((i<n && j<m && a[i]<b[j]) || (j>=m))
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p++] = b[j++];
23.
24.}
```



### Ý TƯỞNG THUẬT TOÁN MERGE SORT

# Ý tưởng thuật toán Merge Sort



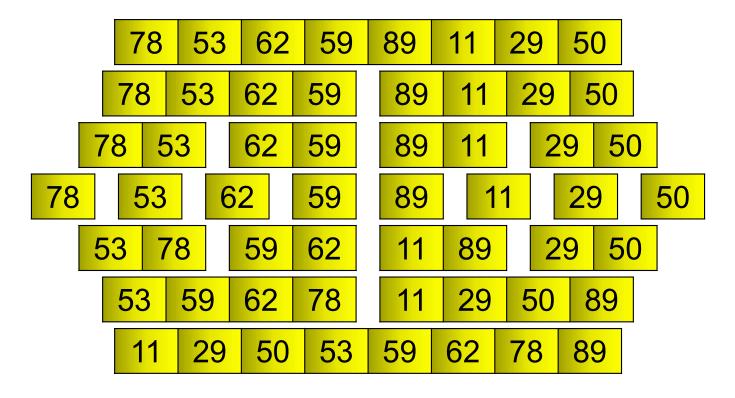
- Thuật toán Merge Sort chia không gian cần sắp xếp thành 2 không gian con.
  - + Nếu không gian con thứ nhất có nhiều hơn một phần tử thì sắp xếp không gian con này bằng thuật toán Merge Sort.
  - + Nếu không gian con thứ hai có nhiều hơn một phần tử thì sắp xếp không gian con này bằng thuật toán Merge Sort.
- Trộn 2 không gian con đã được sắp xếp lại với nhau.



### CHẠY TỪNG BƯỚC THUẬT TOÁN

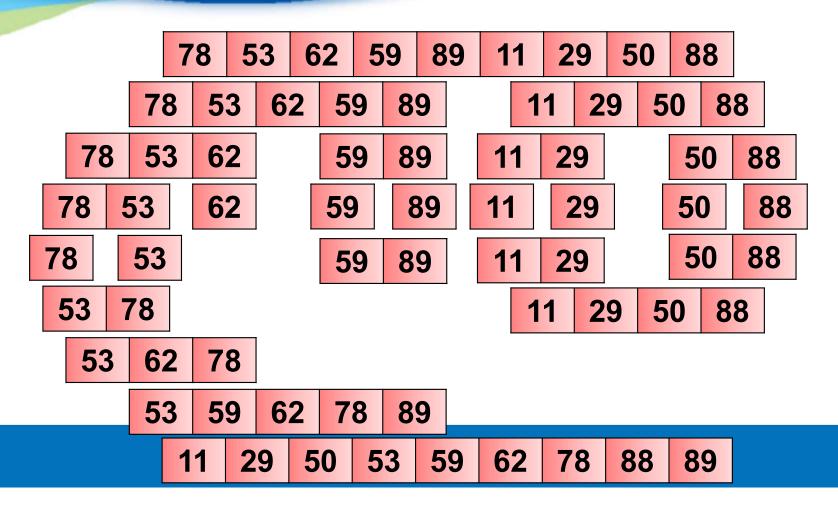
# Chạy từng bước thuật toán





## Chạy từng bước thuật toán







## VÀI CÂU HỎI GỢI Ý CHO HÀM CÀI ĐẶT

# Vài câu hỏi gợi ý cho hàm cài đặt



- Trong đoạn [0 ... 3] có bao nhiều phần tử?
- Trả lời: có 4 phần tử.
- Trong đoạn [5 ... 5] có bao nhiều phần tử?
- Trả lời: có 1 phần tử.
- Trong đoạn [5 ... 9] có bao nhiều phần tử?
- Trả lời: có 5 phần tử.
- Trong đoạn [left ... right] có bao nhiều phần tử?
- Trả lời: có right left + 1 phần tử.



# HÀM CÀI ĐẶT



```
11.void SapTang(int a[], int n)
12.{
13. | MergeSort(a,0,n-1);
14.}
```





```
11.void MergeSort(int a[],int Left,int Right)
12.{
13.
       if(Left<Right)//Đoạn cần sắp có nhiều hơn 1 phần tử
14.
15.
           int Mid = (Left+Right)/2;
16.
           MergeSort(a, Left, Mid);
                                       //Sắp xếp bên trái
                                       //Sắp xếp bên phải
17.
           MergeSort(a,Mid+1,Right);
18.
           Merge(a, Left, Mid, Right);
                                       //Trộn
19.
20.}
```



```
11.void Merge(int a[], int Left, int Mid,int Right)
12.{
13. I
     int* b = new int[Mid-Left+1];
                                    //Left..Mid
14.
     int k;
     15.
     int 1;
16.
17.
     int* temp = new int[Right-Left+1]; //Left..Right
18.
     int m;
19. l
```





```
11.void Merge(int a[], int Left, int Mid, int Right)
12.{
13.
14.
       Tron(b,k,c,l,temp,m);
15.
       for(int i=0; i<m; i++)
16.
           a[Left+i] = temp[i];
       delete []b;
17.
                                  2
                                      3
       delete []c;
18.
                          16
                              22
                                  41
                                     43
                                         56
                                             66
                                                 81
                                                    99
19.
       delete []temp;
                          Left=8
20.}
        3
                5
                           8
                               9
                                      11
                                         12 13 14
                                                    15
            4
                   6
                                  10
                                                        16
                          22
                              66
                                  81
                                     99
                                         16
                                             41
                                                 43
                                                    56
```

```
11.void Tron(int a[],int n,int b[],int m,
             int c[],int &p)
12.
13.{
14.
       int i = 0;
15.
       int j = 0;
16.
     p = 0;
17.
       while(!(i>=n \&\& j>=m))
18.
19.
           if((i<n && j<m && a[i]<b[j]) || (j>=m))
20.
               c[p++] = a[i++];
21.
           else
22.
               c[p++] = b[j++];
23.
24.}
```



#### **CẢI TIẾN HÀM MERGE**

# Cải tiến hàm Merge



```
11.void Merge(int a[], int Left,int Mid,int Right)
12.{
13.     int* temp = new int[Right-Left+1];
14.     int m;
15.     int i = Left; //Left...Mid
16.     int j = Mid+1; //Mid+1...Right
```





```
11.void Merge(int a[], int Left,int Mid,int Right)
12.{
13.
14.
       m = 0;
15.
       while(!(i>Mid && j>Right))
16.
            if( (i<=Mid&&j<=Right&&a[i]<a[j]) ||</pre>
                                                    j>Right)
17.
18.
                temp[m++] = a[i++];
19.
            else
20.
                temp[m++] = a[j++];
21.
22.
```

## Cải tiến hàm Merge

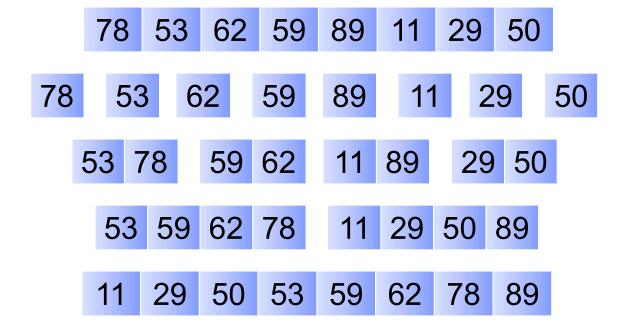


```
11.void Merge(int a[], int Left,int Mid,int Right)
  12.{
  13.
  14.
         for(i=0; i<m; i++)
  15. |
             a[Left+i] = temp[i];
  16.
         delete []temp;
  17.}
                                   2 3 4 5
                            16
                               22
                                  41
                                      43
                                         56
                                             66
                                                81
                                                    99
                           Left=8
          3
                 5
                    6
                            8
                                  10
                                      11
                                         12 13 14 15 16
0
      2
              4
                           22
                               66
                                  81
                                      99
                                         16
                                             41
                                                43
                                                    56
```



## KHỬ ĐỆ QUY HÀM CÀI ĐẶT







78	53	62	59	89	11	29	50
78	53	62	59	89	11	29	50
53	78	62	59	89	11	29	50
53	78	59	62	89	11	29	50
53	78	59	62	11	89	29	50
53	78	59	62	11	89	29	50

53	78	59	62	11	89	29	50
53	59	62	78	11	89	29	50
53	59	62	78	11	29	50	89
11	29	50	53	59	62	78	89





18	29	98	56	78	28	16	57	99	81	22	66	16	41
18	29	98	56	78	28	16	57	99	81	22	66	16	41
18	29	56	98	28	78	16	57	81	99	22	66	16	41
18	29	56	98	16	28	57	78	22	66	81	99	16	41
16	18	28	29	56	57	78	98	16	22	41	66	81	99
16	16	18	22	28	29	41	56	57	66	78	81	98	99





















```
11. void Merge2Run(int a[], int n, int vt, int l)
  12.{
  13.
          int* temp = new int[2*1];
  14.
          int k;
  15.
          int i = vt; //vt ... vt+(1-1)
  16.
          int j = vt+1; //vt+(1-1)+1 ... vt+(1-1)+1+(1-1)
  17.
  18.
          k = 0;
  19.
                            a[], n=18, vt=8, l=4
                             8
                                           12 13 14 15 16
       2
           3
                 5 6
                                 9
                                    10
                                       11
                                                             17
              4
   29
      98
          56
              78
                 28
                     16
                        57
                            99
                                81
                                   22
                                       66
                                           16 41
                                                  56
                                                      43
                                                         22
                                                             78
18
```

0





```
11. void Merge2Run(int a[], int n, int vt, int l)
12.{
13.
14.
       while(!(!(i<vt+1&&i<n) && !(j<vt+2*1&&j<n)))
15.
            if((i<vt+1 && i<n && j<vt+2*1 && j<n &&
16.
                a[i]<a[j]) || !(j<vt+2*1 && j<n))</pre>
17.
18.
                temp[k++] = a[i++];
19.
            else
20.
                temp[k++] = a[j++];
21.
22.
```

11.void Merge2Run(int a[],int n,int vt,int l)



```
12.{
   13.
          for(i=0; i<k; i++)
   14.
               a[vt+i] = temp[i];
   15.
   16.
          delete []temp;
                                           3
                                                  5
                                                          7
                               0
   17.}
                               16
                                  22
                                      41
                                          43
                                              56
                                                  66
                                                      81
                                                          99
                              vt=8,1=4
                               8
                                   9
                                      10
                                          11
                                              12
                                                  13 14
                                                         15
0
        2
           3
               4
                   5
                       6
                                                             16
                                                                 17
18
   29
       98
           56
               78
                   28
                       16
                          57
                              22
                                  66
                                      81
                                          99
                                              16
                                                  41
                                                      43
                                                         56
                                                             22
                                                                 78
```



#### Thuật toán Merge sort

### ĐẶC ĐIỂM – ĐIỂM MẠNH – ĐIỂM YẾU

# Đ<mark>ặc điểm – điểm</mark> mạnh – điểm yếu



- Đặc điểm thuật toán Merge sort:
  - + Độ phức tạp về thời gian (time complexity):  $O(n \log(n))$ .
  - + Độ phức tạp về bộ nhớ (space complexity): O(n).
  - + Trường hợp xấu nhất (worst case):  $O(n \log(n))$ .
  - + Trường hợp trung bình (average case):  $O(n \log(n))$ .
  - + Trường hợp tốt nhất (best case): $O(n \log(n))$ .
  - + Thuộc họ giải thuật chia để trị (Divide and Conquer).
  - + Ôn định.

# Đ<mark>ặc điểm – điểm</mark> mạnh – điểm yếu



- Điểm mạnh:
  - + Hiệu quả trên dữ liệu lớn.
  - + Hiệu suất thuật toán là tối ưu.
  - + Thuật toán thực hiện nhanh.
  - + Thuật toán có thể chạy song song.





# Đặc điểm – điểm mạnh – điểm yếu 🖥



#### – Điểm yếu:

+ Yêu cầu dung lượng bộ nhớ lớn.



#### Cảm ơn quí vị đã lắng nghe

#### ĐẠI HỌC QUỐC GIA TP.HCM TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM TOÀN DIỆN - SÁNG TẠO - PHỤNG SỰ