

2. Ý TƯỞNG THUẬT TOÁN

- 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.

3. ỨNG DỤNG THUẬT TOÁN

78 53 62 59 89 11 29 50

78 53 62 59 89 11 29 50

78 53 62 59 89 11 29 50

78 53 62 59 89 11 29 50

53 78 59 62 11 89 29 50

53 59 62 78 11 29 50 89

11 29 50 53 59 62 78 89

THUẬT TOÁN MERGE SORT

- Bài toán dẫn nhập: 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 dụ:

Mảng a

15	27	32	64
----	----	----	----

Mảng b

-5	1	18	29	37	56	80	93
----	---	----	----	----	----	----	----

Mảng c

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

1. BÀI TOÁN DẪN NHẬP

```
1. void Tron(int a[],int n,  
             int b[],int m,  
             int c[],int &p)  
2. {  
3.     int i = 0;  
4.     int j = 0;  
5.     p = 0;  
6.     while(! (i>=n && j>=m) )  
7.     {  
8.         if( (i<n&&j<m&&a[i]<b[j]) || (j>=m) )  
9.             c[p++] = a[i++];  
10.        else  
11.            c[p++] = b[j++];  
12.    }  
13. }
```

4. HÀM CÀI ĐẶT

```
1. void MergeSort(int a[], int Left, int Right)
2. {
3.     if(Left<Right)
4.         //Mảng có nhiều hơn 1 phần tử
5.         {
6.             int Mid = (Left+Right)/2;
7.             // Sắp xếp mảng bên trái
8.             MergeSort(a,Left,Mid);
9.             // Sắp xếp mảng bên phải
10.            MergeSort(a,Mid+1,Right);
11.            // Trộn 2 mảng lại với nhau
12.            Merge(a,Left,Mid,Right);
13.        }
14. }
```

4. HÀM CÀI ĐẶT

```
1. void Merge(int a[], int Left,int Mid,int Right)
2. {
3.     int b[100];int k;
4.     int c[100];int l;
5.     int temp[100];int n;
6.     k = 0;
7.     for(int i=Left;i<=Mid;i++)
8.         b[k++] = a[i];
9.     l = 0;
10.    for(int i=Mid+1;i<=Right;i++)
11.        c[l++] = a[i];
12.    Tron(b,k,c,l,temp,n);
13.    for(int i=0;i<n;i++)
14.        a[Left+i] = temp[i];
15. }
```

4. HÀM CÀI ĐẶT

```
1. void Tron(int a[],int n,  
            int b[],int m,  
            int c[],int &p)  
2. {  
3.     int i = 0;  
4.     int j = 0;  
5.     p = 0;  
6.     while(!(i>=n && j>=m))  
7.     {  
8.         if((i<n&&j<m&&a[i]<b[j]) || (j>=m))  
9.             c[p++] = a[i++];  
10.        else  
11.            c[p++] = b[j++];  
12.    }  
13. }
```

4. HÀM CÀI ĐẶT

```
1. void SapTang(int a[], int n)
2. {
3.     MergeSort(a, 0, n-1);
4. }
```