

## BÀI 10: GIẢI THUẬT SẮP XẾP NÂNG CAO

### Bài 1: Sắp xếp phân đoạn – Quick sort

Bài toán:

Cho dãy số nguyên  $x = \{34, 14, 24, 54, 84, 64, 94, 74, 04\}$

Yêu cầu:

- Mô tả quá trình sắp xếp dãy số nguyên  $x$  theo chiều giảm dần bằng phương pháp sắp xếp PHÂN ĐOẠN.
- Thiết kế giải thuật sắp xếp.
- Cài đặt chương trình ứng dụng.

#### Hướng dẫn thực hành

- Mô tả quá trình sắp xếp phân đoạn

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
	[34]	14	24	54	(84)	64	94	74	(04)
	[34]	14	24	(54)	(04)	64	(74)	(94)	(84)
	[34]	(14)	24	(04)	(54)	(64)	74	[84]	[94]
	(04)	[14]	(24)	(34)	[54]	[64]	[74]		
			[24]	[34]					
K.Quả	04	14	24	34	54	64	74	84	94

- Giải thuật sắp xếp phân đoạn

```
void quick_sort(int x[], int left, int right){
    if (left < right) {
        int k = (left + right) / 2, t = x[k];
        int i = left, j = right;
        do{
            while (x[i] < t) i = i + 1;
            while (x[j] > t) j = j - 1;
            if (i <= j) {
                int tg = x[i];
                x[i] = x[j]; x[j] = tg;
                i = i+1; j = j-1;
            }
        }
```

```

    }
    while (i <= j);
    quick_sort(x, left, j);
    quick_sort(x, i, right);
}
}
- Ứng dụng giải thuật
int main() {
    int x[9] = {34, 14, 24, 54, 84, 64, 94, 74, 04};
    quick_sort(x, 0, 8);
    for(int i = 0; i < 9; i++)
        cout<<x[i]<<" ";
    return 0;
}

```

## ***Bài 2: Sắp xếp vun đống – Heap sort***

Bài toán: Cho dãy số nguyên x = {14, 24, 54, 84, 64, 94, 74}

Yêu cầu:

- Mô tả quá trình sắp xếp dãy số nguyên x theo chiều giảm dần bằng phương pháp sắp xếp vun đống.
- Thiết kế giải thuật sắp xếp.
- Cài đặt chương trình ứng dụng.

### ***Hướng dẫn thực hành***

- ***Mô tả quá trình sắp xếp***

Lượt vun đống thứ nhất

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 1	34	14	24	<u>54</u>	84	64	94	<u>74</u>	<u>04</u>
	34	14	<u>24</u>	<b>74</b>	84	<u>64</u>	<u>94</u>	<b>54</b>	04
	34	<u>14</u>	<b>94</b>	<u>74</u>	<u>84</u>	64	<b>24</b>	54	04
	<u>34</u>	<u>84</u>	<u>94</u>	74	<b>14</b>	64	24	54	04
	<b>94</b>	84	<u>34</u>	74	14	<u>64</u>	<u>24</u>	54	04
	<u>94</u>	84	<b>64</b>	74	14	<b>34</b>	24	54	<u>04</u>
Đổi chỗ phần tử đầu và cuối	04	84	64	74	14	34	24	54	<u>94</u>

Lượt vun đống thứ hai

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 2	<u>04</u>	<u>84</u>	<u>64</u>	74	14	34	24	54	<u>94</u>
	<b>84</b>	<u>04</u>	64	<u>74</u>	<u>14</u>	34	24	54	
	84	<u>74</u>	64	<u>04</u>	14	34	24	<u>54</u>	
	<b>84</b>	74	64	<b>54</b>	14	34	24	<u>04</u>	
Đổi chỗ phần tử đầu và cuối	04	74	64	54	14	34	24	<u>84</u>	

Lượt vun đồng thứ ba

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 3	<u>04</u>	<u>74</u>	<u>64</u>	54	14	34	24	<u>84</u>	<u>94</u>
	<b>74</b>	<u>04</u>	64	<u>54</u>	<u>14</u>	34	24		
	<u>74</u>	<b>54</b>	64	<b>04</b>	14	34	<u>24</u>		
Đổi chỗ phần tử đầu và cuối	24	54	64	04	14	34	<u>74</u>		

Lượt vun đồng thứ tư

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 4	<u>24</u>	<u>54</u>	<u>64</u>	04	14	34	<u>74</u>	<u>84</u>	<u>94</u>
	<b>64</b>	54	<u>24</u>	04	14	<u>34</u>			
	<b>64</b>	54	<b>34</b>	04	14	<u>24</u>			
Đổi chỗ phần tử đầu và cuối	24	54	34	04	14	<u>64</u>			

Lượt vun đồng thứ năm

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 5	<u>24</u>	<u>54</u>	<u>34</u>	04	14	<u>64</u>	<u>74</u>	<u>84</u>	<u>94</u>
	<b>54</b>	<u>24</u>	34	<u>04</u>	<u>14</u>				
	<u>54</u>	24	34	04	<u>14</u>				
Đổi chỗ phần tử đầu và cuối	14	24	34	04	<u>54</u>				

Lượt vun đồng thứ sáu

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 6	<u>14</u>	<u>24</u>	<u>34</u>	04	(54)	(64)	(74)	(84)	(94)
	<u>34</u>	24	<b>14</b>	<u>04</u>					
Đổi chỗ phần tử đầu và cuối	04	24	14	(34)					

Lượt vun đồng thứ bảy

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 7	<u>04</u>	<u>24</u>	<u>14</u>	(34)	(54)	(64)	(74)	(84)	(94)
	<u>24</u>	<b>04</b>	<u>14</u>						
Đổi chỗ phần tử đầu và cuối	14	04	(24)						

Lượt vun đồng thứ tám

	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]
Lần 8	<u>14</u>	<u>04</u>	(24)	(34)	(54)	(64)	(74)	(84)	(94)
	14	04							
Đổi chỗ phần tử đầu và cuối	04	(14)							
K.Quả	<b>04</b>	<b>14</b>	<b>24</b>	<b>34</b>	<b>54</b>	<b>64</b>	<b>74</b>	<b>84</b>	<b>94</b>

Giải thuật sắp xếp vun đồng

### 1. Vun đồng cho một nút

```
void vun_dong(int x[], int k, int n) {
    if (k <= n/2 - 1) {
        int j = 2 * k + 1;
        if (j < n-1 && x[j] < x[j+1])
            j = j + 1;
        if (x[k] < x[j]) {
            int tg = x[k];
            x[k] = x[j];
            x[j] = tg;
            vun_dong(x, j, n);
        }
    }
}
```

```
}
```

## 2. Tạo đống đầu tiên

```
void tao_dong_dau_tien(int x[], int n)
{
    for (int k = n/2 - 1; k >= 0; k--)
        vun_dong(x, k, n);
}
```

## 3. Giải thuật sắp xếp vun đống

```
void heapSort(int x[], int n)
{
    tao_dong_dau_tien (x, n);
    for (int i = n; i >= 2; i--)
    {
        int tg = x[0];
        x[0] = x[i-1];
        x[i-1] = tg;
        vun_dong(x, 0, i-1);
    }
}
```

### ***Bài 3: Sắp xếp trộn – Merge sort***

Bài toán:

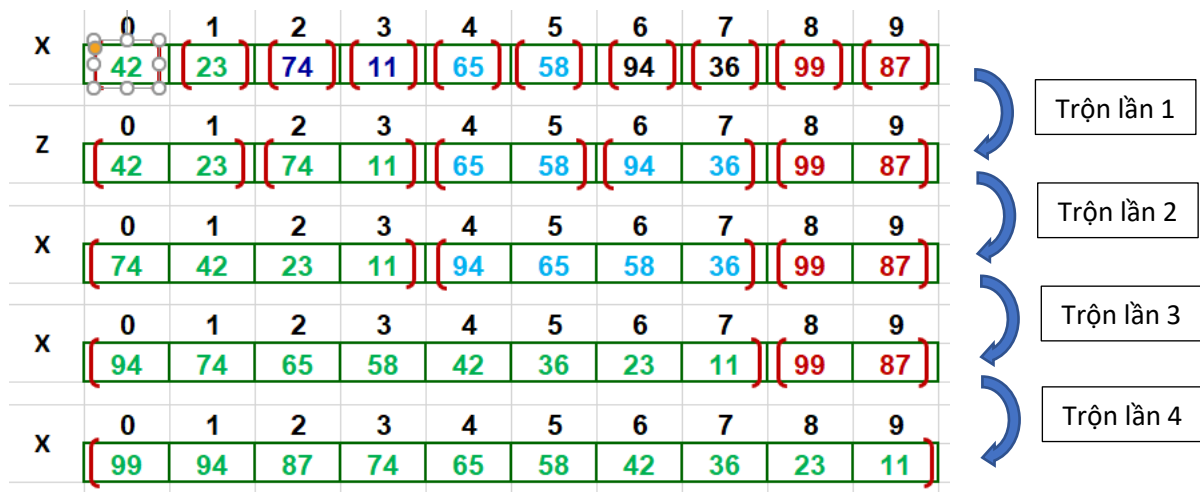
Cho dãy số nguyên  $x = \{42, 23, 74, 11, 65, 58, 94, 36, 99, 87\}$

Yêu cầu:

- Mô tả quá trình sắp xếp dãy số nguyên x theo chiều giảm dần bằng phương pháp sắp xếp trộn.
- Thiết kế giải thuật sắp xếp.
- Cài đặt chương trình ứng dụng.

Hướng dẫn

#### 1. Mô tả quá trình sắp xếp



## 2. Giải thuật sắp xếp trộn

Trộn hai vệt thành một vệt:

```
void merge(int *X,int bt1,int w1,int bt2,int w2,int *Z)
{
```

*//bt1, bt2: là vị trí biên trái của hai vệt, w1, w2 là độ dài của hai vệt*

```
    int i = bt1, j = bt2, bp1 = bt1 + w1 - 1;
```

```
    int bp2 = bt2 + w2 - 1, k = bt1;
```

*//bp1, bp2 là biên phải của hai vệt, k là biên trái của vệt mới trên Z*

```
    while (i <= bp1 && j <= bp2) {
```

```
        if (X[i] < X[j]) {
```

```
            Z[k] = X[i]; i++; k++;
```

```
        }
```

```
        else {
```

```
            Z[k] = X[j]; j++; k++;
```

```
        }
```

```
    }
```

```
    while (i <= bp1) {
```

```
        Z[k] = X[i]; i++; k++;
```

```
    }
```

```
    while (j <= bp2) {
```

```
        Z[k] = X[j]; j++; k++;
```

```
    }
```

```
}
```

Trộn các cặp vệt trên dãy X, chuyển sang dãy Z

```
void mergePass (int X[],int n,int K,int Z[]) {
```

*//Z là dãy chứa dãy X sau khi trộn các cặp vệt*

*//1. Khởi tạo các giá trị ban đầu*

```
int cv = n / (2 * K); //Số cặp vệt  
int s = 2 * K * cv; //Số pt có cặp độ dài K  
int r = n - s; //Số pt lẻ cặp
```

*//2. Trộn từng cặp vệt*

```
for (int j = 1; j <= cv; j++){  
    b1 = (2 * j - 2) * K; //biên trái của vệt thứ nhất  
    merge(X, b1, K, b1 + K, K, Z);  
}
```

*//3. Chỉ còn một vệt*

```
if (r <= K)  
    for (int j=0; j<r; j++) {  
        Z[s + j] = X[s + j];  
    }
```

*//4. Còn hai vệt nhưng một vệt có độ dài nhỏ hơn K*

```
else merge(X, s, K, s + K, r - K, Z);  
}
```

**Giải thuật sắp xếp trộn**

```
void mergeSort (int X[], int n)  
{  
    //1. Khởi tạo số phần tử trong một vệt  
    int K = 1;  
    //2. Sắp xếp trộn  
    while (K < n)  
    {  
        //Trộn và chuyển các phần tử vào dãy Z  
        mergePass(X, n, K, Z);  
        //Trộn và chuyển các phần tử trở lại dãy X  
        mergePass(Z, n, 2 * K, X);  
        K = K * 2;  
    }  
}
```

**Bài 4: Ứng dụng sắp xếp danh sách**

**Yêu cầu:**

- Tạo một danh sách học sinh, mỗi học sinh gồm các thông tin: họ và tên, giới tính, năm sinh, điểm tổng kết.
- Sử dụng giải thuật sắp xếp phân đoạn để sắp xếp danh sách theo tên với thứ tự từ điển. Hiện thị danh sách được sắp xếp.
- Sử dụng giải thuật sắp xếp vun đống để sắp xếp danh sách theo thứ tự giảm dần của năm sinh. Hiện thị danh sách được sắp xếp.
- Sử dụng giải thuật sắp xếp trộn để sắp xếp danh sách theo thứ tự tăng dần của điểm tổng kết. Hiện thị danh sách được sắp xếp.

## **BÀI TẬP VỀ NHÀ**

### ***Bài 1: Sắp xếp phân đoạn – Quick sort***

Bài toán: Cho dãy từ {John, Try, Thor, Zil, Adam, Dany, Milk}

Yêu cầu:

- Mô tả quá trình sắp xếp dãy từ trên theo thứ tự từ điển bằng phương pháp sắp xếp phân đoạn.
- Thiết kế giải thuật sắp xếp
- Cài đặt chương trình ứng dụng.

### ***Bài 2: Sắp xếp vun đống – Heap sort***

Bài toán: Cho dãy từ {John, Cool, Bee, Try, Thor, Zil, Adam }

Yêu cầu:

- Mô tả quá trình sắp xếp dãy từ trên theo thứ tự từ điển bằng phương pháp sắp xếp lựa chọn.
- Thiết kế giải thuật sắp xếp.
- Cài đặt chương trình ứng dụng.

### ***Bài 3: Sắp xếp trộn – Merge sort***

Bài toán: Cho dãy từ {John, Cool, Bee, Try, Thor, Zil, Adam, Dany, Milk}

Yêu cầu:

- Mô tả quá trình sắp xếp dãy từ trên theo thứ tự từ điển bằng phương pháp sắp xếp chèn.
- Thiết kế giải thuật sắp xếp.
- Cài đặt chương trình ứng dụng.