

BÀI 02

CHIẾN LƯỢC ĐỆ QUY – QUAY LUI – CHIA ĐỂ TRỊ

Design by Minh An

Email: anvanminh.hau@gmail.com

1

Nội dung

- ❖ Bài toán liệt kê
- ❖ Một số kiến thức về đại số tổ hợp
- ❖ Phương pháp sinh
- ❖ Đệ quy
- ❖ Quay lui
- ❖ Chia để trị
- ❖ Một số bài tập

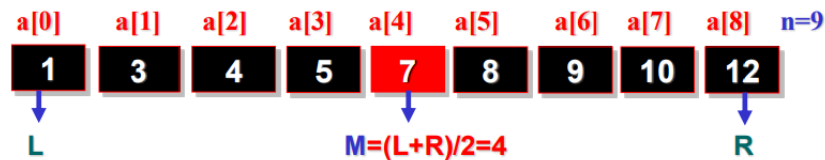
Design by Minh An

2

Chiến lược chia để trị

1. TÌM KIẾM NHỊ PHÂN

Cho một dãy a gồm n phần tử đã được sắp tăng, cho một phần tử C . Cho biết C có xuất hiện trong a hay không?



C=10

```
a[M]=c:    Yes
a[M] < c:    L=M+1;
a[M] > c:    R=M-1;
L>R        :    No
```

Design by Minh An

3

Chiến lược chia để trị

1. TÌM KIẾM NHỊ PHÂN

```
a[M]=c :    Yes
a[M] < c:    L=M+1;
a[M] > c:    R=M-1;
L>R     :    No
```

```
int TKNP_DQ(int a[100], int c, int L, int R)
{
    int M=(L+R)/2;

    if(suy_biến)
        return <CÔNG_THỨC_SUY_BIẾN>;
    else
        return <CÔNG_THỨC_TỔNG_QUÁT>;
}
```

Design by Minh An

4

Chiến lược chia để trị

1. TÌM KIẾM NHỊ PHÂN

| | | |
|------------|---|---------------|
| $a[M] = c$ | : | Yes |
| $a[M] < c$ | : | $L = M + 1$; |
| $a[M] > c$ | : | $R = M - 1$; |
| $L > R$ | : | No |

```
int TKNP_Lap(int a[100], int n, int c)
{
    int L=0, R=n-1, M;
    do
    {
        M = (L+R)/2;
        if (a[M]>c) R=M-1;
        if (a[M]<c) L=M+1;
    }
    while(a[M]!=c && L<R);
    if (a[M]==c) return M;
    else return -1;
}
```

Design by Minh An

5

Chiến lược chia để trị

1. TÌM KIẾM NHỊ PHÂN

Cho một dãy a gồm n phần tử đã được sắp tăng, cho một phần tử C . Cho biết C có xuất hiện trong a hay không?

Nếu C không xuất hiện trong a , hãy tìm vị trí để chèn C vào a mà không phá vỡ tính được sắp của a .

```
for i = 1 to n do
    if A[i] ≥ q then
        return index i
return n + 1
```

```
Search(a, L, R, C)
    if L = R then
        return L (index)
    M = (L + R) / 2
    if C < a[M] then
        return Search(a, L, M, C)
    else
        return Search(a, M, R, C)
```

Design by Minh An

6

Chiến lược chia để trị

2. CHIẾN THUẬT CHIA ĐỂ TRỊ

- A top-down technique for designing algorithms
- **Dividing** the problem into smaller subproblems
- Hoping that the solutions of the sub-problems are easier to find
- **Composing** the partial solutions into the solution of the original problem

Design by Minh An

7

Chiến lược chia để trị

2. CHIẾN THUẬT CHIA ĐỂ TRỊ

Divide-and-conquer paradigm consists of following major phases:

- **Breaking** the problem into several sub-problems that are similar to the original problem but smaller in size.
- **Solve** the sub-problem recursively (successively and independently), and then
- **Combine** these solutions to subproblems to create a solution to the original problem.

Design by Minh An

8

Chiến lược chia để trị

3. SẮP XẾP TRỘN

Bài toán trộn:

- Cho hai dãy các phần tử cùng kiểu đã được sắp tăng, hãy trộn hai dãy để thu được một dãy cũng được sắp tăng.

$i=0 \rightarrow$

a

| | | |
|---|---|---|
| 1 | 3 | 5 |
|---|---|---|

 $n=3$

$j=0 \rightarrow$

b

| | | | | | |
|---|---|---|---|---|---|
| 2 | 4 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|

 $m=6$

$k=0 \rightarrow$

c

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|

Design by Minh An

9

Chiến lược chia để trị

3. SẮP XẾP TRỘN

Bài toán trộn:

$i=0 \rightarrow$

a

| | | | |
|---|---|---|----|
| 1 | 3 | 5 | 10 |
|---|---|---|----|

 $n=3$

$j=0 \rightarrow$

b

| | | | | | | |
|---|---|---|---|---|---|----|
| 2 | 4 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|----|

 $m=6$

$k=0 \rightarrow$

c

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|

Design by Minh An

10

Chiến lược chia để trị

3. SẮP XẾP TRỘN

Bài toán trộn:

```
int c[100];
void Tron2(int a[50], int n, int b[50], int m)
{
    int Max=a[n-1];
    if (Max<b[m-1]) Max=b[m-1];
    a[n]=b[m]=Max+1;
    //-----
    int i=0, j=0;
    for(int k=0; k<n+m; k++)
        if (a[i]<b[j])
            {c[k]=a[i]; i++;}
        else
            {c[k]=b[j]; j++;}
}
```

Design by Minh An

11

Chiến lược chia để trị

3. SẮP XẾP TRỘN

Bài toán trộn:

$i=0 \rightarrow$

$\leftarrow j=n-1$

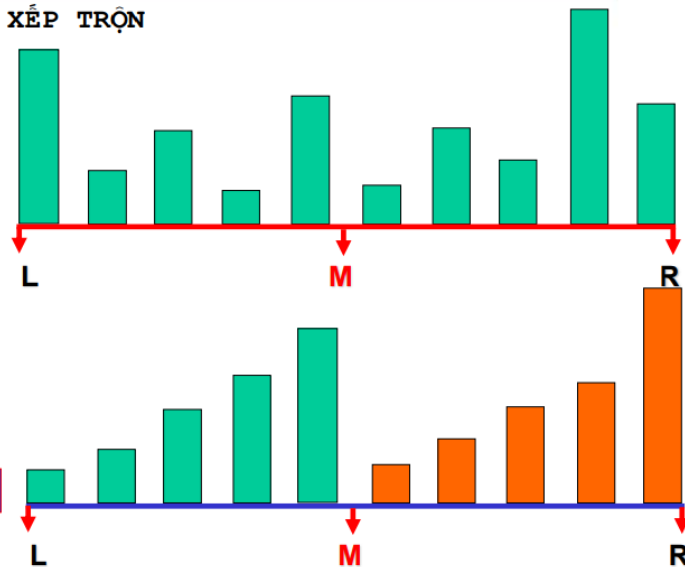
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 9 | 8 | 7 | 6 | 4 | 2 |
|---|---|---|---|---|---|---|---|---|

Design by Minh An

12

Chiến lược chia để trị

3. SẮP XẾP TRỘN

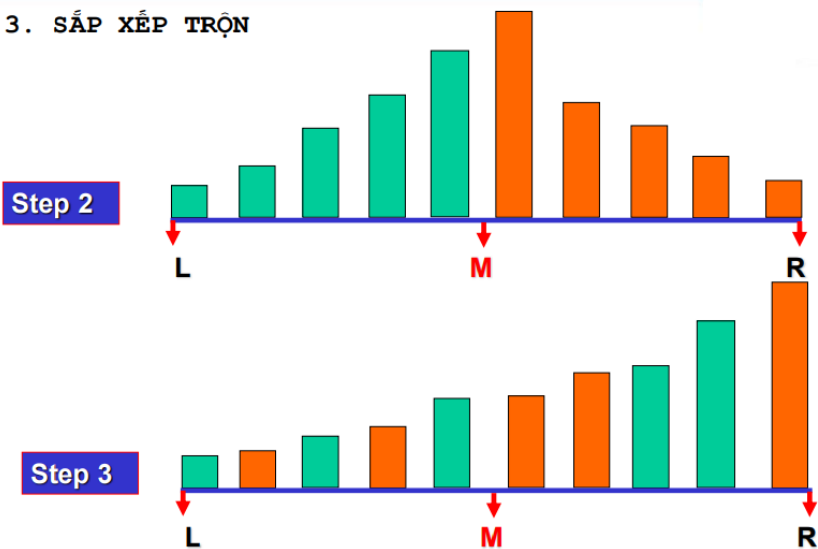


Design by Minh An

13

Chiến lược chia để trị

3. SẮP XẾP TRỘN



Design by Minh An

14

Chiến lược chia để trị

3. SẮP XẾP TRỘN

```
void MergeSort(float a[], int l, int r)
{
    if(r>l)
    {
        int m=(l+r)/2;
        MergeSort(a,l,m); MergeSort(a,m+1, r);
        - Sao chép nửa đầu của a sang b
        - Sao chép nửa còn lại sang b theo thứ tự ngược lại
        - Trộn hai nửa.
    }
}
```

Design by Minh An

15

Chiến lược chia để trị

3. SẮP XẾP TRỘN

```
void MergeSort(float a[], int l, int r)
{
    if(r>l)
    {
        int m=(l+r)/2;
        MergeSort(a,l,m); MergeSort(a,m+1, r);
        //Sao chép nửa đầu của a sang b
        for(int i=m; i>=l; i--) b[i]=a[i];
        //Sao chép nửa còn lại của a sang b theo thu tu nguoc lai
        for(int j=m+1; j<=r;j++) b[r+m+1-j]=a[j];
        //i chạy từ đầu mảng b, j chạy từ cuối mảng b và trộn
        i=l; j=r;
        for(int k=l; k<=r; k++)
            if(b[i]<b[j]) {a[k]=b[i];i++;}
            else {a[k]=b[j];j--;}
    }
}
```

Design by Minh An

16

Chiến lược chia để trị

4. TÌM MAX

Cho một dãy số thực gồm n phần tử, hãy tìm phần tử lớn nhất của dãy.



MAX = 9



a = 9

b=7

Design by Minh An

17

Chiến lược chia để trị

Bài 5: Tính a^n bằng chiến lược chia để trị (a thực, n nguyên dương).

Bài 6: Tính tổng các số lẻ trong một mảng n số nguyên bằng phương pháp chia để trị. Áp dụng tính giá trị trung bình cộng các số lẻ (nếu có).

Bài 7: Công ty vận tải A có n chiếc xe tải với tải trọng khác nhau, thông tin về mỗi xe tải gồm: biển số và tải trọng. Viết chương trình thực hiện:

- Khởi tạo danh sách gồm ít nhất 7 xe tải.
- Xác định xe tải có tải trọng lớn nhất bằng chiến lược chia để trị.
- Sắp xếp danh sách xe tải theo thứ tự tăng dần của tải trọng bằng thuật toán trộn.
- Tìm vị trí để chèn một xe tải mới vào danh sách sao cho trật tự danh sách không bị thay đổi.

Design by Minh An

18