

BÀI 02

CHIẾN LƯỢC ĐỆ QUY – QUAY LUI – CHIA ĐỂ TRỊ

Design by Minh An

Email: anvanminh.hau@gmail.com

1

Nội dung

- ❖ Phương pháp sinh
- ❖ Giải thuật đệ quy
- ❖ Chiến lược quay lui
- ❖ **Chiến lược chia để trị**
- ❖ Một số bài tập

Design by Minh An

2

3.5. Chiến lược chia để trị

3.5.1. Tìm kiếm nhị phân

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



21

Design by Minh An

3

Tìm kiếm nhị phân

Thuật toán tìm kiếm nhị phân theo phương pháp đệ quy

```
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

Tìm kiếm nhị phân

Thuật toán tìm kiếm nhị phân theo phương pháp lặp

```
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

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

3.5.2. Khái niệm chia để trị

- ✓ Là một chiến lược thiết kế thuật toán theo kiểu top-down
- ✓ Chia bài toán thành các bài toán cùng dạng nhưng có kích thước nhỏ hơn
- ✓ Hi vọng rằng việc tìm kiếm lời giải của các bài toán con là đơn giản hơn
- ✓ Tổng hợp lời giải của các bài toán con để được lời giải của bài toán ban đầu

Design by Minh An

7

Khái niệm chia để trị

- Chiến lược chia để trị giải quyết bài toán qua 3 bước
 - ✓ **Chia** bài toán thành các bài toán con cùng dạng nhưng kích thước nhỏ hơn
 - ✓ Giải quyết (**trị**) các bài toán con (theo cách đệ quy)
 - ✓ **Tổng hợp lời giải** của các bài toán con để được lời giải của bài toán ban đầu

Design by Minh An

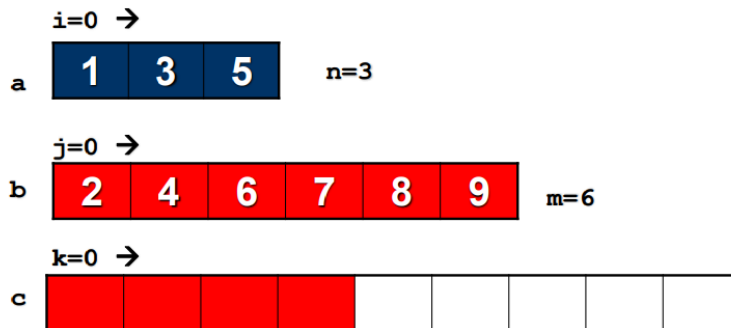
8

3.5.3. Giải quyết bài toán bằng chiến lược chia để trị

3.5.3.1. Thuật toán sắp xếp trộn

▪ Bài toán trộn

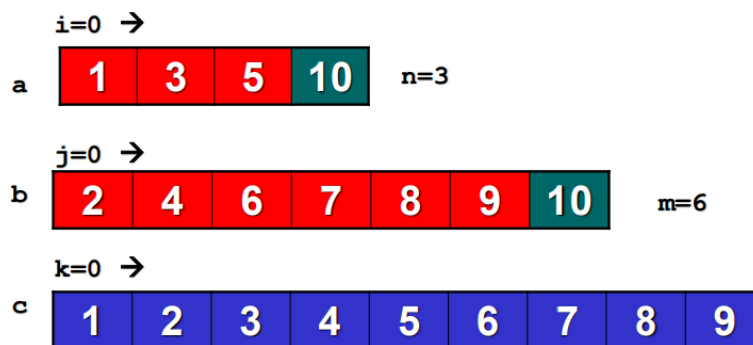
- ✓ Cho 2 dãy a và b gồm các phần tử cùng kiểu dữ liệu được sắp tăng, hãy trộn hai dãy đã cho để được một dãy c cũng được sắp tăng.



Design by Minh An

9

Bài toán trộn



Design by Minh An

10

Bài toán trộn

▪ Thuật 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

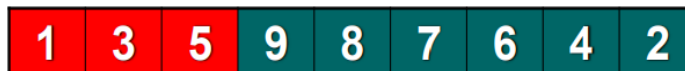
Thuật toán sắp xếp trộn

▪ Ý tưởng của thuật toán sắp xếp trộn

- ✓ Chia để trị
- ✓ Trộn hai dãy đã sắp

i=0 →

← j=n-1

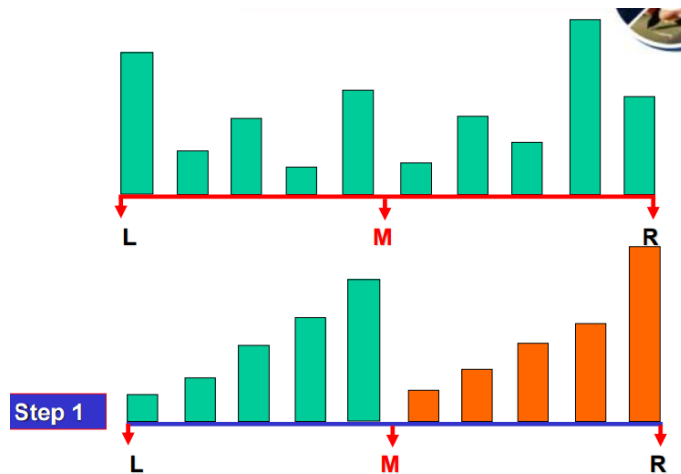


Design by Minh An

12

Thuật toán sắp xếp trộn

- **Chia:** Chia dãy cần sắp thành 2 dãy con, **đễ**
- **Trị:** Sắp xếp 2 dãy con theo thứ tự tăng

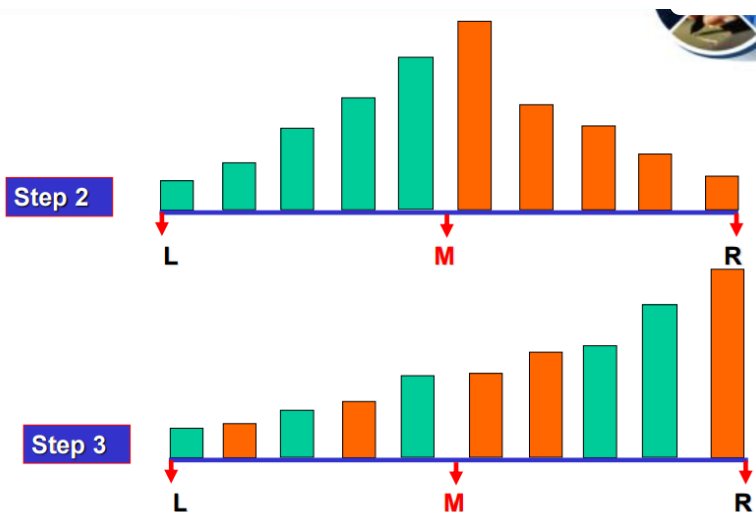


Design by Minh An

13

Chiến lược chia để trị

- **Tổng hợp lời giải:** Trộn 2 dãy con được dãy cần sắp



Design by Minh An

14

Thuật toán 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

Thuật toán 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 thứ tự ngược lại
        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

3.5.3.2. Tìm max: Cho một dãy số thực a gồm n phần tử, hãy tìm phần tử lớn nhất của dãy.



MAX = 9



a = 9

b=7

Design by Minh An

17

Bài tập

1. Tính a^n bằng chiến lược chia để trị (a thực, n nguyên dương).
2. Thiết kế thuật toán 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).
3. 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