



CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Giảng viên: Dương Trần Đức

Khoa: Công nghệ thông tin I

Học viện Công nghệ Bưu chính Viễn thông



ĐỆ QUI CHIA ĐỂ TRỊ (DIVIDE AND CONQUER)

- Giới thiệu
- Ví dụ
- Bài tập



Giải thuật đệ qui chia để trị

- Chia bài toán ban đầu thành 2 hoặc nhiều bài toán con tương tự
- Giải quyết lần lượt từng bài toán con này
 - Bài toán con là dạng đơn giản hơn -> gọi đệ qui
- Thông thường:
 - Chia tập dữ liệu ban đầu thành 2
 - Gọi 2 thủ tục đệ qui cho 2 phần vừa chia



Mã giả hàm đệ quy chia để trị

```
void DAC(A, x){  
    if (A nhỏ) Solve(A)  
    else{  
        Chia A thành các bài toán con  $A_1,$   
         $A_2, \dots A_k$   
        For  $i=1$  to  $k$  DAC( $A_i, x_i$ )  
        Kết hợp  $x_i$  ( $i=1, \dots, k$ ) thành nghiệm  
        bài toán A là x  
    }  
}
```

VD1: Tìm giá trị lớn nhất trong dãy

- Đơn giản: Duyệt và so sánh
- Đệ qui:
 - Chia dãy làm 2 nửa
 - Tìm giá trị lớn nhất của mỗi nửa và so sánh với nhau

```
double max(double a[], int l, int r) {  
    if (l==r) return a[l];  
    m=(l+r)/2;  
    double u=max(a,l,m)  
    double v=max(a,m+1,r)  
    if (u>v) return u; else return v;  
}
```



VD2: Sắp xếp nhanh (QuickSort)

- Sắp 1 dãy tăng dần

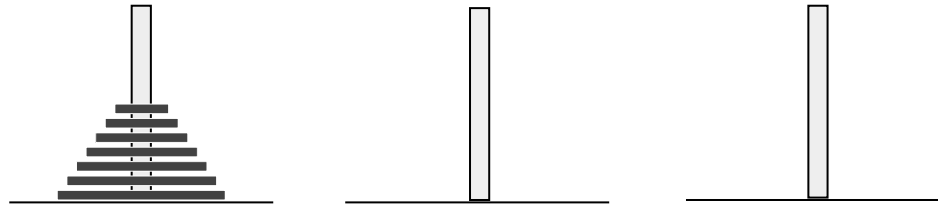
- Chia dãy thành 2 nửa: Các số nửa đầu $<$ các số nửa sau
- Tiến hành sắp xếp trên mỗi nửa
- Ghép 2 nửa dãy sau khi sắp xếp



Mã giả QuickSort

```
void QS(A, left, right){
    i=left;j=right;
    x=(left+right)/2;
    do{
        while(A[i]<x && i<right) i++;
        while(A[j]>x && j>left) j--;
        if (i<=j){swap(A[i],A[j]);i++;j--}
    }while(i<=j)
    if (left<j) QS(A,left,j);
    if (i<right) QS(A,i,right);
}
```

VD3: Bài toán Tháp Hà nội (Hanoi Tower)



- Yêu cầu: Chuyển bộ n đĩa từ cọc ban đầu A sang cọc đích C (có thể sử dụng cọc trung gian B)
 - Mỗi lần chuyển 1 đĩa
 - Đĩa có kích thước nhỏ hơn bao giờ cũng phải nằm trên đĩa có kích thước lớn hơn



VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- $N=1$

- Chuyển trực tiếp

- $N=2$

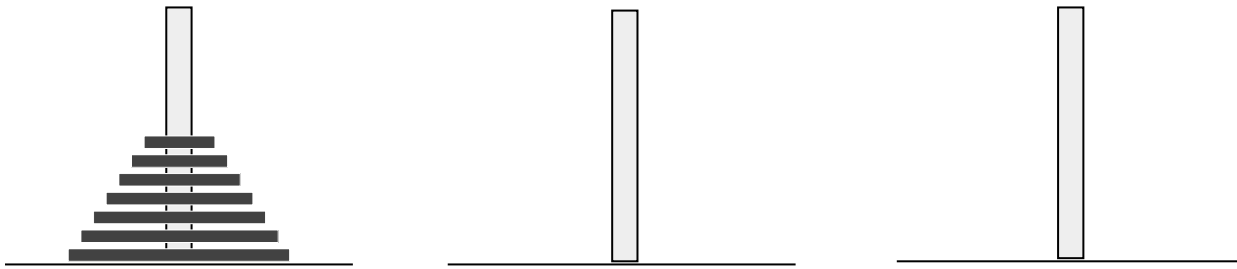
- Chuyển đĩa nhỏ từ cọc A sang cọc trung gian B.
- Chuyển đĩa lớn từ cọc A sang cọc đích C.
- Chuyển đĩa nhỏ từ cọc trung gian B sang cọc đích C.

- $N>2$

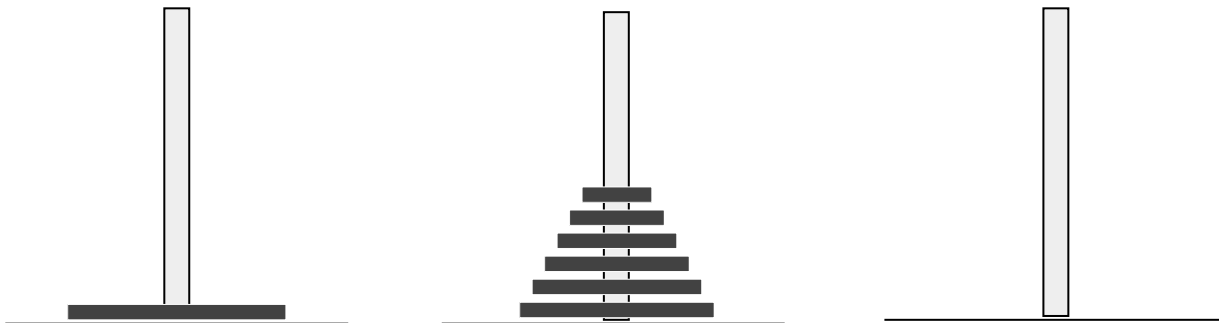
- Chuyển $n-1$ đĩa bên trên sang cọc trung gian B.
- Chuyển đĩa dưới cùng (đĩa thứ n) sang cọc đích C.
- Chuyển $n-1$ đĩa từ cọc trung gian B sang cọc đích C.

VD3: Bài toán Tháp Hà nội (Hanoi Tower)

○ Trạng thái ban đầu:

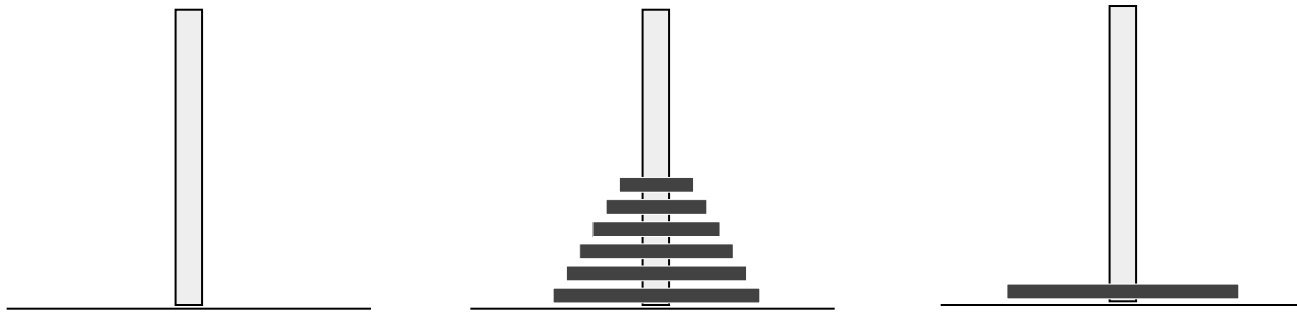


○ Bước 1: Chuyển $n-1$ đĩa bên trên từ cọc A sang cọc B, sử dụng cọc C làm cọc trung gian

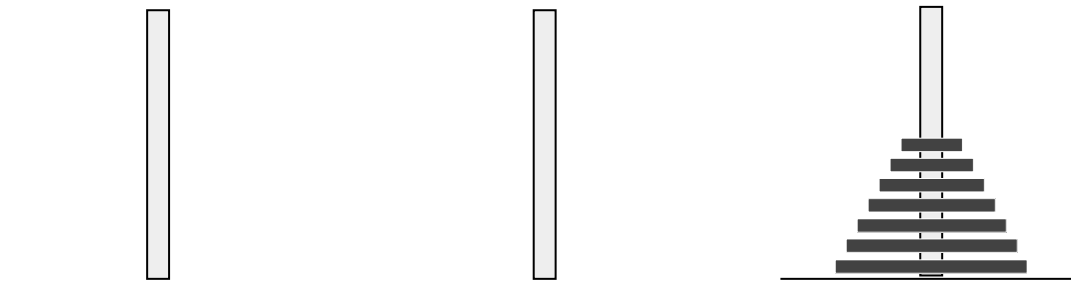


VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- Bước 2: Chuyển đĩa cuối cùng từ cọc A sang cọc C



- Bước 3: Chuyển $n-1$ đĩa từ cọc B sang cọc C sử dụng cọc A làm cọc trung gian



- ● ● | VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- Bài toán Tháp Hà nội: $N=3$



- ● ● | VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- Bài toán Tháp Hà nội: $N=4$





VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- Chia để trị: Bài toán chuyển n đĩa được chia làm 2 bài toán nhỏ hơn là chuyển $n-1$ đĩa
- Cài đặt đệ qui:
 - Hàm `chuyen(int n, int a, int c)` thực hiện việc chuyển đĩa thứ n từ cọc a sang cọc c .
 - Hàm `thaphanoi(int n, int a, int c, int b)` là hàm đệ qui thực hiện việc chuyển n đĩa từ cọc a sang cọc c , sử dụng cọc trung gian là cọc b .



VD3: Bài toán Tháp Hà nội (Hanoi Tower)

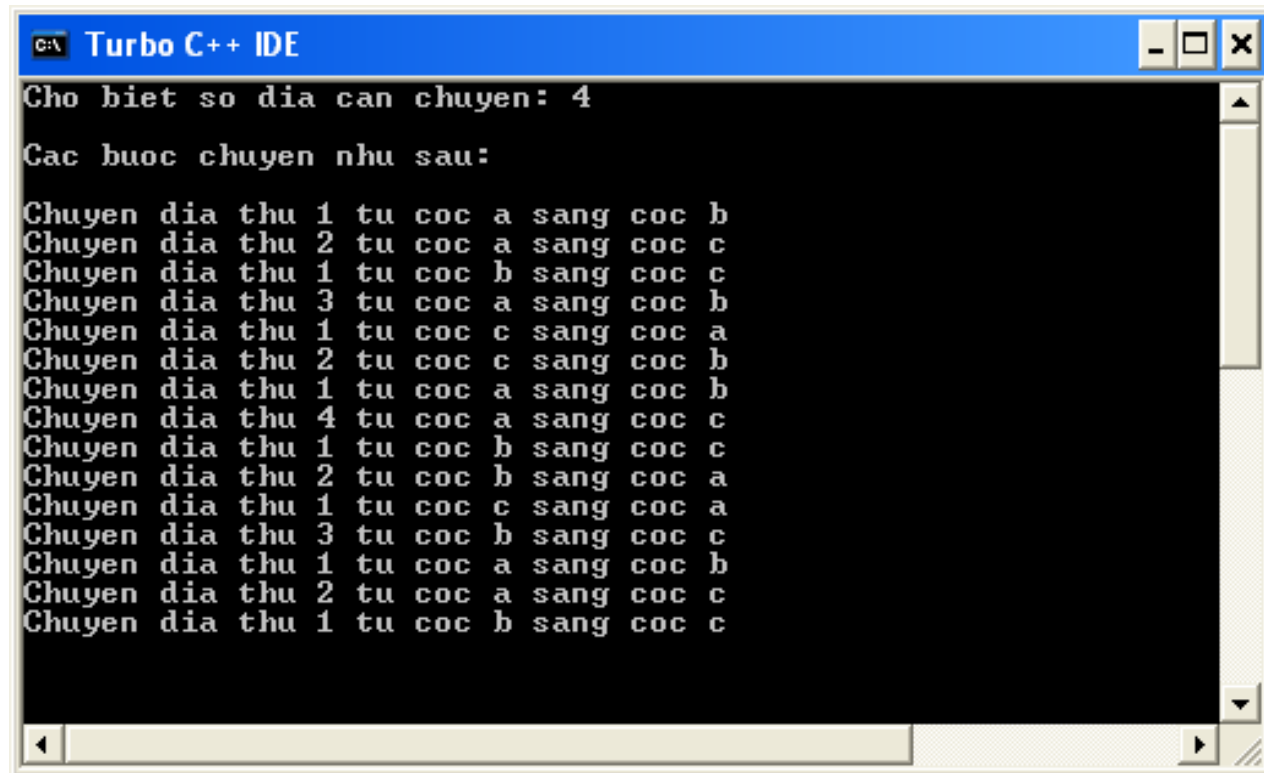
```
void chuyen(int n, char a, char c){
    printf('Chuyen dia thu %d tu coc %c sang coc %c\n',n,a,c);
    return;
}

void thaphanoi(int n, char a, char c, char b){
    if (n==1) chuyen(1, a, c);
    else{
        thaphanoi(n-1, a, b, c);
        chuyen(n, a, c);
        thaphanoi(n-1, b, c,a);
    }
    return;
}
```

VD3: Bài toán Tháp Hà nội (Hanoi Tower)

- Độ phức tạp của thuật toán:

- Cấp độ tăng: $2^n - 1$ (chứng minh bằng qui nạp)
- 40 đĩa: 34.800 năm! (mỗi lần chuyển 1 giây)



```
C:\ Turbo C++ IDE
Cho biet so dia can chuyen: 4
Cac buoc chuyen nhu sau:
Chuyen dia thu 1 tu coc a sang coc b
Chuyen dia thu 2 tu coc a sang coc c
Chuyen dia thu 1 tu coc b sang coc c
Chuyen dia thu 3 tu coc a sang coc b
Chuyen dia thu 1 tu coc c sang coc a
Chuyen dia thu 2 tu coc c sang coc b
Chuyen dia thu 1 tu coc a sang coc b
Chuyen dia thu 4 tu coc a sang coc c
Chuyen dia thu 1 tu coc b sang coc c
Chuyen dia thu 2 tu coc b sang coc a
Chuyen dia thu 1 tu coc c sang coc a
Chuyen dia thu 3 tu coc b sang coc c
Chuyen dia thu 1 tu coc a sang coc b
Chuyen dia thu 2 tu coc a sang coc c
Chuyen dia thu 1 tu coc b sang coc c
```




Bài tập

- Viết chương trình thực hiện các VD bằng C/C++
- Tính A^n bằng thuật toán chia để trị
 - Gợi ý: $A^n = A^{n/2} \times A^{n/2}$ nếu n chẵn và $= A \times A^{n/2} \times A^{n/2}$ nếu n lẻ
- Thực hiện tìm kiếm phần tử q trong mảng A bằng tìm kiếm nhị phân
 - Sắp xếp dãy A
 - So sánh q với phần tử ở giữa dãy mid
 - Nếu $q > mid$ tìm kiếm ở nửa sau dãy
 - Nếu $q < mid$ tìm kiếm nửa đầu dãy