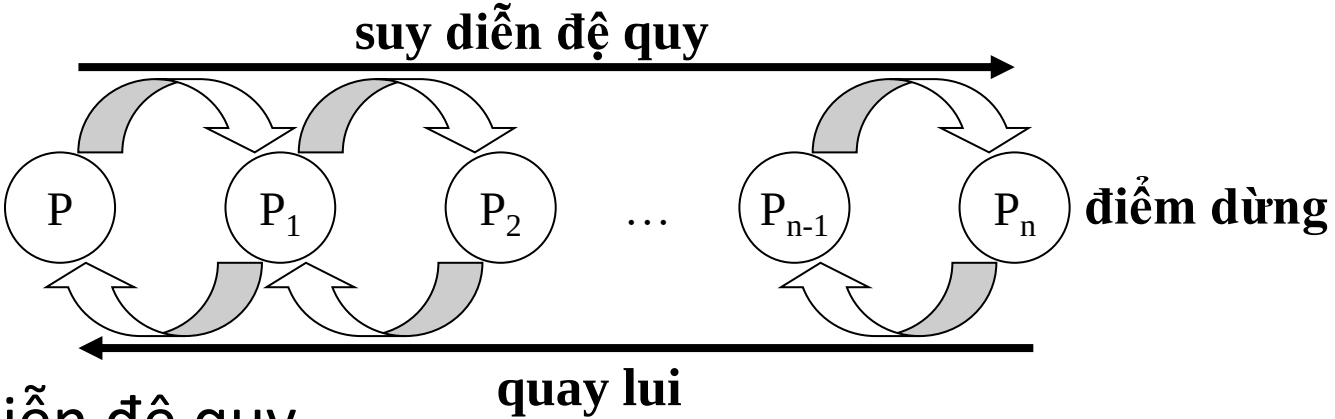
Chương 3: Đệ quy

Data structures and Algorithms

Nội dung chính

- Giải thuật đệ quy
  - Cấu tạo giải thuật đệ quy
  - Hoạt động của giải thuật đệ quy
- Sự khử đệ quy
- Giải thuật quay lui

Giải thuật đệ quy



- Quá trình suy diễn đệ quy
  - Đưa về P một bài toán P_1 có bản chất tương tự P nhưng có quy mô bé hơn.
  - Đưa P_1 về bài toán P_2 có cùng bản chất với P_1 và cũng có quy mô nhỏ hơn P_1.
  - Quá trình cứ tiếp tục cho đến khi đưa bài toán về bài toán con P_n tương tự như P_{n-1} và có quy mô nhỏ hơn P_{n-1}.
  - Pn có thể giải một cách trực tiếp (điểm dừng)
- Quá trình quay lui
  - Sau khi giải được Pn, ta quay lại giải các bài toán con theo trật tự ngược lại và cuối cùng giải được bài toán ban đầu P.

Giải thuật đệ quy

- Ví dụ: hàm tính giai thừa một số nguyên không âm

Quy ước:

1. n = 0 thì n! = 1;
2. n > 0 thì n! = n(n-1)!

Hàm tính n!

Giải thuật

```
Function giaiThua(n)
    If n = 0 then FACT = 1;
    Else FACT = n*FACT(n-1)
    Return
```

Cài đặt

```
long giaiThua(int n)
{
   if (n==0) return 1;
   else return   n*giaiThua(n-1);
}
```

T(n) = O(n)

Dãy fibonacci

- Dãy số Fibonacci là dãy số có dạng như sau:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55 …

- Định nghĩa Fib(n) như sau

1. Nếu n = 1 hoặc n = 2 thì Fib(n) = 1
2. Nếu n >2 thì Fib(n) = Fib(n-1) + Fib(n-2)

Giải thuật tìm số thứ n trong dãy Fibonacci

Function FIB(n)

1. If n==0 or n==1 then FIB = 1;
   Else FIB = FIB(n-1) + FIB(n-2)
2. Return

Dãy Fibonacci

Đưa ra phần tử thứ n – có đệ quy

```
#include<stdio.h>
#include<conio.h>
int Fib(int i)
{
      if(i == 0||i==1)
      {
            return 1;
      }
      return Fib(i-1) + Fib(i-2);
}
```

```
int  main()
{
      int i, n ;
      printf(”Nhap so phan tu trong day Fibonacci: ");
      scanf_s("%d",&n);
      for (i = 0; i < n; i++)
      {
       printf("%d\t", Fib(i));
       }
      return 0;
getch();
}
```

Sự khử đệ quy

- Khi thay các giải thuật đệ quy bằng các giải thuật không tự gọi chúng, ta gọi đó là ***sự khử đệ quy***
- Sự khử đệ quy được thực hiện thông qua các vòng lặp (for, while) và phân nhánh (if...then...else)
- Ví dụ 1:

```
int giaiThua (int n){
    if  (n <= 1) return 1;
    else {
        int gt=1;
        for (int i=2;i<=n;i++)
            gt=gt*i;
        return gt;
    }
}
```

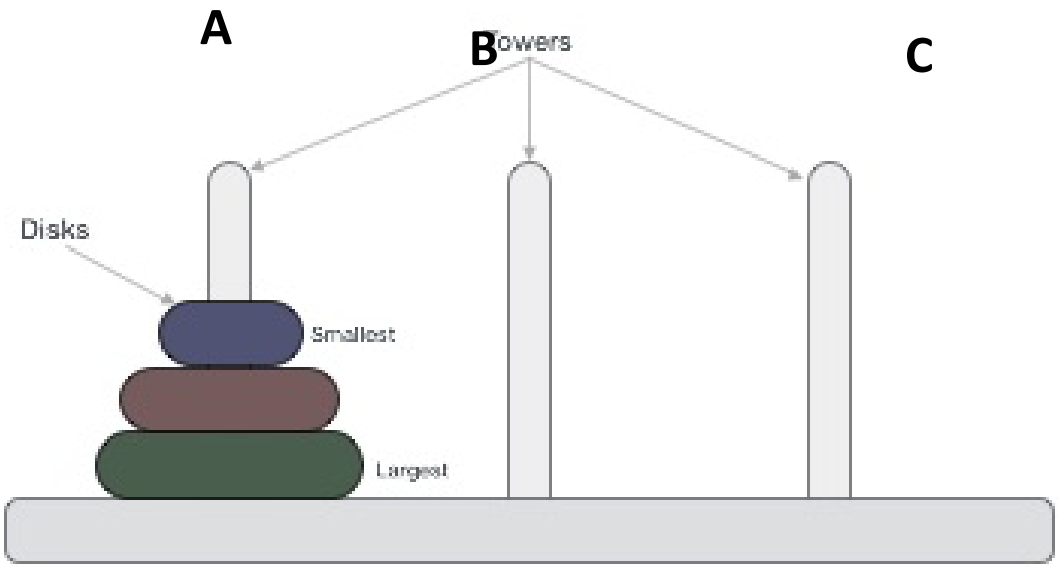
Dãy Fibonacci

đưa ra phần tử thứ n trong dãy Fibonacci – khử đệ quy

```
int Fib(int n) {
   int f1 = 1,  f2 = 1;
   int f;
   int i;
   if (n == 1 || n == 2)
         return 1;
   for (i = 3; i <= n; i++) {
         f = f1 + f2;
        f1 = f2;
         f2 = f;
  }
   return f;
}
```

```
void main() {
   int i;
   printf("10 so dau tien cua day
Fibonacci: \n");
   for (i = 1; i < =10; i++) {
      printf("%d ", Fib(i));
   }
}
```

Bài toán tháp Hà nội



- Có 3 tháp A, B, C trong đó tháp A có N đĩa kích thước khác nhau, được xếp lên nhau theo thứ tự đĩa nhỏ hơn được đặt chồng lên đĩa lớn hơn. Thực hiện chuyển N đĩa từ tháp A sang tháp B với điều kiện:
  - Mỗi lần chỉ chuyển một đĩa
  - Không có tình huống đĩa to ở trên đĩa nhỏ ở dưới (dù là tạm thời)
  - Được phép sử dụng một cọc trung gian

Bài toán tháp Hà nội

- Nếu N=1: chuyển đĩa từ cọc A sang cọc B
- Trái lại, thực hiện theo các bước sau:
  - Chuyển N-1 đĩa từ A sang C với B làm trung gian
  - Chuyển 1 đĩa (là đĩa lớn nhất) từ A sang B
  - Chuyển N-1 đĩa từ C sang B, với A làm trung gian

Bài toán tháp Hà nội

```
Procedure Tower(n,A,B,C)
If n=1 then chuyen dia tu A sang B
Else begin
    call Tower(n-1,A,C,B);
    call Tower(1,A,B,C);
    call Tower(n-1,C,B,A)
    end
End.
```

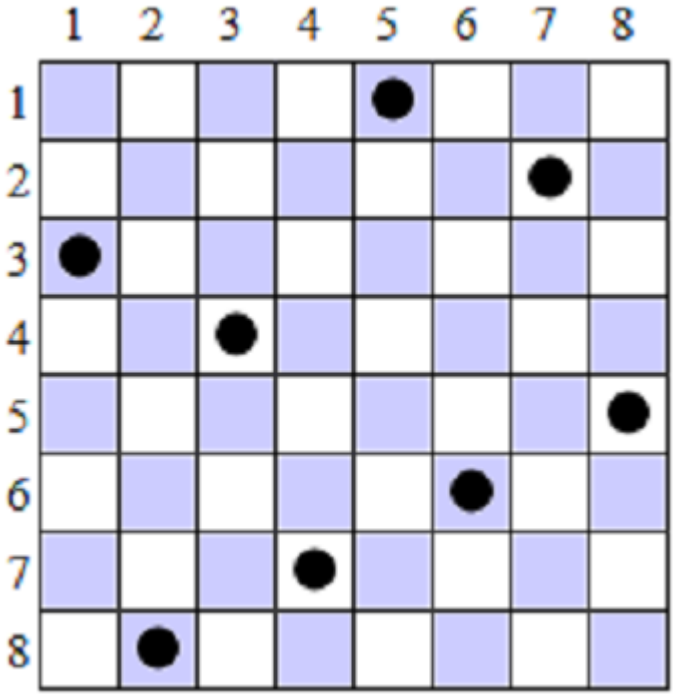
Bài toán tháp Hà nội

```
void Tower(int n , char a, char b, char c ){
   if(n==1){
      printf("\t%c-------%c\n",a,c);
      return;
   }
   Tower(n-1,a,c,b);
   Tower(1,a,b,c);
   Tower(n-1,c,b,a);
   }
```

```
int main(){
   char a='A', b='B', c='C';
   int n;
   printf("Nhap n: ");
   scanf("%d",&n);
   Tower(n,a,b,c);
}
```

Giải thuật quay lui

- Là một dạng của giải thuật đệ quy
- Ví dụ bài toán 8 con hậu: Hãy tìm cách bố trí 8 con hậu trên bàn cờ sao cho không có hai con hậu nào ăn được nhau. Bài toán này có thể mở rộng cho N con hậu



Giải thuật quay lui

- Ý tưởng giải thuật:
  - Coi bàn cờ là một mảng hai chiều kích thước NxN
  - Gọi N con hậu lần lượt là h_1, h_2 …, $h_{N.}$
  - Để các con hậu không ăn nhau, ta đặt con hậu thứ i (h_i) vào hàng thứ i, với i=1…N.
  - Chọn vị trí cột thích hợp cho từng con hậu để chúng không ăn nhau

Giải thuật quay lui

- Chi tiết giải thuật:
  - Giả sử ở bước thứ i, ta đã xếp được i con hậu h_1, h_2 …, h_i vào i hang đầu tiên, nếu i bằng N thì kết thúc, đưa ra kết quả. Trái lại thì sang bước tiếp theo
  - Tại bước thứ i+1, tìm vị trí cột thích hợp ở hàng i+1 để đưa h_{i+1} vào đó. Có hai khả năng:
    - Nếu tìm được vị trí cột j thích hợp (không bị một trong i con hậu đầu tiên ăn) thì đưa h_{i+1} vào vị trí j.
    - Nếu cả N cột đều không thích hợp thì quay lại thử tiếp ở bước thứ I (quay lui) để tìm vị trí thích hợp tiếp theo của hi (hiện tại hi đang ở vị trí thích hợp). Nếu không có vị trí tiếp theo thích hợp và I bằng 1 thì giải thuật cũng kết thúc với kết luận không có giải pháp nào