

## Bài tập thuật toán đệ quy

***Yêu cầu:***

- 1. Trình bày thuật toán đệ quy giải các bài toán sau theo hướng dẫn.***
- 2. Thực thi các thuật toán theo input cô đã cho và bôi vàng.***  
***(đối với những bài tập cô gọi ý thuật toán rồi các em chỉ cần thực thi thuật toán thôi)***

**Bài 1. Tính  $C_n^k$  theo công thức đệ quy:  $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$ ,**

**Trường hợp dừng:  $k = 0$  hoặc  $k = n$  thì  $C_n^k = 1$ .**

***Hướng dẫn***

*Function tohop(n, k):*

- Trường hợp cơ sở:  $k=0$  hoặc  $k=n$
- Trường hợp tổng quát:  $tohop(n-1, k) + tohop(n-1, k-1)$

***(Thực thi thuật toán với  $n=5, k=3$ )***

***{nội dung kết quả thực thi ở đây}***

**Bài 2. Đếm số chữ số của một số nguyên dương n.**

***Hướng dẫn***

*Function demchuso(n):*

- Trường hợp cơ sở:  $n \leq 9$ : số có 1 chữ số
- Trường hợp tổng quát:  $demchuso(n/10)$

***Thực thi thuật toán với  $n=9872$***

**Bài 3. Kiểm tra một mảng số nguyên có phải mảng đối xứng không?**

***Hướng dẫn***

*Function ktdoixung(a[], dau, cuoi):*

- Trường hợp cơ sở: chỉ số đầu tiên của mảng lớn hơn hoặc bằng chỉ số cuối cùng của mảng ( $dau \geq cuoi$ )
- Trường hợp tổng quát:  $ktdoixung(a, dau+1, cuoi-1)$

(Kiểm tra điều kiện: phần tử đầu khác với phần tử cuối của mảng **đang xét** không, nếu có thì không đối xứng)

### **Thuật toán đệ quy**

Function ktdoixung(a[], dau, cuoi):

if (dau >= cuoi): return true end if

if (a[dau] != a[cuoi]): return false end if

return ktdoixung(a, dau+1, cuoi-1)

end if

end function

**Thực thi thuật toán với a[] = {5 3 2 2 3 5} và a[] = {4 5 6 3 4}**

**Bài 4. Tìm số lớn nhất trong mảng số nguyên.**

### **Hướng dẫn**

- Trường hợp cơ sở: mảng có 1 phần tử khi đó số lớn nhất chính là a[0]

- Trường hợp tổng quát: tìm phần tử lớn nhất của mảng có n-1 phần tử max(a, n-1)

+ Kiểm tra điều kiện nếu phần tử cuối cùng của mảng lớn hơn phần tử lớn nhất của mảng gồm (n-1) phần tử còn lại thì trả về phần tử cuối cùng của mảng, ngược lại trả về phần tử lớn nhất của mảng có (n-1) phần tử còn lại.

### **Thuật toán đệ quy**

Function max(a[], n):

if (n == 1): return a[0] end if

if (a[n-1] > max(a, n-1)):

return a[n-1]

else return max(a, n-1)

end if

end function

**Thực thi thuật toán với n=7, a[] = {15 3 2 12 30 5 10}**

**Bài 5. Tính n!!**

### **Hướng dẫn**

*Function giaithuakep(n):*

- Trường hợp cơ sở: ( $n=0$  ||  $n=1$ ): kết quả bằng 1
- Trường hợp tổng quát: kết quả bằng  $n \cdot \text{giaithuakep}(n-2)$

**Thực thi thuật toán với  $n=7$  và  $n=10$**

**Bài 6. Kiểm tra một số nguyên dương  $n$  có phải là số nguyên tố không?**

**Hướng dẫn**

- Trường hợp cơ sở: kiểm tra tất cả các ước của số  $n$  và không tìm thấy ước nào nằm trong khoảng từ 2 đến căn bậc hai của  $n$ .
- Trường hợp tổng quát: kiểm tra  $n$  có chia hết cho các số từ 2 đến căn bậc hai của  $n$  giảm đi 1 số  $i$  ( $\text{nguyento}(n, i+1)$ )

**Thuật toán đệ quy**

*Function nguyento(n, i):*

```
if (n<2): return false end if
if (n==2): return true end if
if (i>sqrt(n)): return true end if
if (n%i==0): return false end if
return nguyento(n,i+1)
end function
```

**Thực thi thuật toán với  $n=17$  và  $n=8$**

**Bài 7. Kiểm tra chuỗi  $s1$  có phải là chuỗi con của chuỗi  $s2$  không?**

Một chuỗi  $s1$  được coi là chuỗi con của chuỗi  $s2$  khi mà mọi ký tự của chuỗi  $s1$  xuất hiện trong chuỗi  $s2$  theo thứ tự, nhưng không nhất thiết phải liên tiếp.

**Hướng dẫn**

- Trường hợp cơ sở: khi một trong hai chuỗi rỗng ( $m=0$  hoặc  $n=0$ )
- Trường hợp tổng quát: so sánh 2 chuỗi với số ký tự mỗi chuỗi giảm đi 1 nếu phần tử cuối cùng của 2 chuỗi bằng nhau ( $s1[m-1]==s2[n-1]$ ) ngược lại so sánh 2 chuỗi với số ký tự của chuỗi  $s1$  giữ nguyên nhưng số ký tự chuỗi  $s2$  giảm đi 1.

**Thuật toán đệ quy**

*Function xaucon(s1,s2,m,n):*

*if (m==0): return true end if*

*if (n==0): return false end if*

*if (s1[m-1]==s2[n-1]):*

*return xaucon(s1,s2,m-1,n-1)*

*else return xaucon(s1,s2,m,n-1)*

*end if*

*end function*

**Thực thi thuật toán với s1="123" và s2="1012534"**

**Bài 8. Kiểm tra mảng có sắp tăng hay không?**

**Hướng dẫn**

- Trường hợp cơ sở: mảng có ít hơn 1 phần tử

- Trường hợp tổng quát: kiểm tra mảng với n-1 phần tử.

+ Điều kiện: Nếu phần tử đứng trước lớn hơn phần tử đứng sau trong mảng thì mảng không tăng.

**Thuật toán đệ quy**

*Function daytang(a[], n):*

*if (n<=1): return true end if*

*if (a[n-1]<a[n-2]): return false end if*

*return daytang(a[],n-1)*

*end function*

**Thực thi thuật toán với n=7, a[]={1 3 5 10 30 33 100}**