

HVIT CLAN JSC

C Programming

Bài toán min-max và sắp xếp

Nội dung slide: Bài toán kinh điển min-max và sorting

Thuyết minh: Nguyễn Đồng Khánh

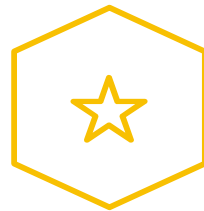
HVIT CLAN 2020



NỘI DUNG CHÍNH (3 nội dung)



**Các mức độ của
giải thuật**



**Bài toán tìm
min/max**



Bài toán sắp xếp

Phần 1

Giải thuật và các
mức độ



Giải thuật và mức độ của giải thuật

Giải thuật: Là cách giải, các bước nhằm giải quyết một bài toán, một vấn đề cụ thể.

Các mức của giải thuật: Được chia theo độ phức tạp của thuật toán.
Như : N , N^2 , N^C , $\log(N)$, $N \cdot \log(N)$...

Nếu có thể luận ngay ra đáp án bằng cách suy nghĩ/nhìn yêu cầu ?

- Nếu vậy, thì luôn tồn tại lời giải THUẦN TÚY NHẤT bằng cách mô phỏng lại đúng những gì đang nghĩ trong đầu ra thành câu lệnh lập trình.
- Sau đó, có thể dùng các nhận xét, chứng minh toán... để làm tối ưu thêm câu lệnh đã viết ra.

Ví dụ: Số đối xứng là số viết xuôi hay viết ngược vẫn là chính nó. Xác định số 1221 có đối xứng không?

- Rõ ràng, ta có thể nhận ngay ra 1221 là đối xứng chỉ bằng cách nhìn nó và suy ngẫm đến định nghĩa của số đối xứng. Nhưng viết ra như thế nào? Muốn vậy, phải suy nghĩ CHẬM LẠI! Não mình đã hoạt động từng bước như thế nào để ta nhận thức ra đây là số đối xứng ?
- Trước hết, não của ta tự “viết” số ngược lại là 1221, rồi so sánh lại với số ban đầu. Giống nhau ! Nên nó kết luận rằng đây là số đối xứng.
- Chỉ là não ta hoạt động quá nhanh, ta không kịp nhận thấy mà thôi!
➔ Vậy thì ta chỉ việc mô phỏng lại não, cách làm thuần túy nhất là tạo ra một số x ngược lại với số ban đầu, rồi so sánh với số ban đầu là xong!

Phần 2

Bài toán tìm min/max



Bài toán tìm min/max

Cho mảng a:

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

Tìm số bé/lớn nhất trong mảng.

Chớp mắt, ta có thể lướt qua và thấy ngay số nhỏ nhất là 1. Vậy nào đã hoạt động thế nào?

*Số X bé nhất, nhìn thấy số nào đầu tiên thì X là số đó!
Giờ thì ta nhìn tiếp từ trái qua phải.
Xem số nào còn bé hơn số X đang nhớ hay không?
Nếu có thì cập nhật lại giá trị mới cho X.
Nếu không thì thôi không làm gì cả, nhìn số tiếp theo.
Nhìn xong hết rồi thì kết luận*

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     //Mảng a có n=8 phần tử
6     int a[10] = { 7,4,3,2,1,8,6,5 }, n = 8;
7     //Số X bé nhất, nhìn thấy số nào đầu tiên thì X là số đó!
8     int min=a[0];
9     //Giờ thì ta nhìn tiếp từ trái qua phải.
10    for (int i=1; i < n; i++)
11    {
12        //Xem số nào còn bé hơn số X đang nhớ hay không?
13        if (a[i] < min)
14            //Nếu có thì cập nhật lại giá trị mới cho X.
15            min = a[i];
16        //Nếu không thì thôi không làm gì cả, nhìn số tiếp theo.
17        else { }
18    }
19    //Nhìn xong hết rồi thì kết luận
20    printf("số bé nhất là %d ", min);
21 }
```

Phần 3

Bài toán sắp xếp



Bài toán sắp xếp

Cho mảng a:

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

Hãy sắp xếp mảng tăng dần/giảm dần.

Có rất nhiều thuật toán sắp xếp,
ta hãy thử phát biểu một thuật toán trong đó:

Bắt đầu từ một phần tử, ta đi so sánh nó với tất cả những phần tử đằng sau nó.

Nếu thấy lớn hơn (hay bé hơn) tùy vào mục đích sắp xếp tăng dần (hay giảm dần), thì đổi chỗ.

Sau khi so sánh xong, lặp lại với phần tử ở vị trí tiếp theo.

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     //Mảng a có n=8 phần tử
6     int a[10] = { 7,4,3,2,1,8,6,5 }, n = 8;
7     //Bắt đầu từ một phần tử (a[0])
8     for (int i = 0; i < n-1; i++)
9     {
10        //Đi so sánh với tất cả phần tử còn lại đằng sau nó
11        // j = i+1 và chạy đến hết
12        for (int j = i + 1; j < n; j++)
13        {
14            //Nếu thấy lớn hơn
15            if (a[i] < a[j])
16            {
17                //Thì đổi chỗ
18                int temp = a[i];
19                a[i] = a[j];
20                a[j] = temp;
21            }
22        }
23        //Kết thúc vòng for con thì số bé nhất đã được đổi về đầu
24        //Ta quay lại với i tăng tiếp lên 1
25        //Tiếp tục lặp lại công việc với phần tử sau phần tử a[0]
26    }
27 }
```


TRÂN TRỌNG CẢM ƠN

