Nhìn nhận Tin học dưới con mắt Toán học

1. sqrt(n) hay n div 2 hay n-1?
2. Tại sao lại là sqrt(n)?

Đối với người mới bắt đầu học Tin học (bao gồm các em học sinh và sinh viên) thì việc tìm hiểu các thuật toán không bao giờ là chuyện dễ dàng cả. Nguyên nhân biện giải cho luận điểm này là:

1. Người thầy còn quá gượng ép học trò trong tiến trình xây dựng thuật toán. Bắt học trò phải xây dựng thuật toán dựa trên ý tưởng của người thầy chứ không tạo điều kiện cho các em tự xây dựng trên nền tảng *"suy nghĩ tự nhiên"* của chính các em.
2. Khi các em chưa nhận thức được vấn đề thì các em buộc phải học thuộc. Học thuộc nhiều rồi cũng thành quen. Và sau đó, dường như các em coi đó như một điều hiển nhiên, không chút ngờ vực. Việc làm này gây tâm lý chán nản khi nhập môn. Đồng thời cũng làm cản trở quá trình sáng tạo trong học tập, khi luôn phải chấp nhận những điều mà mình không hiểu.
3. Nền tảng Toán học của các em còn yếu. Hoặc nguyên nhân cũng do chính người thầy cũng không hiểu tường tận, kỹ càng làm sao lại có thuật toán như vậy, để từ đó giải thích kỹ càng cho học trò của mình.
4. Phải hiểu rằng đủ thời gian thì các em sẽ thành công.
5. Không có gì là ảo thuật khi làm Tin học cả. Làm các em tưởng Tin học là môn học xa rời thực tiễn. Rất nguy hiểm.

Nói thêm rằng: Đối với những người mới nhập môn thì tâm lý làm được bài sẽ quyết định rất nhiều thành công sau này.

Để giải quyết vấn đề này, người viết xin đề xuất một cái nhìn: đó là nhìn nhận Tin học dưới con mắt của người làm Toán. Toán học sẽ là cơ sở soi xét, dẫn dắt, giúp cho Tin học chặt chẽ, rõ ràng và giúp cho học trò dễ dàng tiếp thu bài học.

Mở đầu, bằng một bài toán đã quá quen thuộc. Đó là bài toán: Kiểm tra xem một số nguyên n lớn hơn 1 có phải số nguyên tố hay không?

Có lẽ khi đọc bài này một số người sẽ cho rằng quá dễ. Dễ dàng đến mức hiển nhiên. Và họ viết chương trình ngay như sau:

```
function NguyenTo(n: integer): boolean;
var i: integer;
begin
        NguyenTo:= false;
        for i:= 2 to trunc(sqrt(n)) do
                if n mod i = 0 then exit;
        NguyenTo:= true;
end;
```

Ta thấy chương trình ở trên mục đích là kiểm tra xem ngoài 1 và n xem còn có ước nào khác của n không. Nhưng tại sao lại ta chỉ kiểm tra các số từ 2 đến $\sqrt{n}$ thôi mà không phải từ 2 tới n-1? Bởi theo định nghĩa số nguyên tố thì ta chỉ có: Số nguyên tố là số nguyên lớn hơn 1 và có hai ước là 1 và chính nó.

Với lại tại sao lại tự dưng gán NguyenTo:= false rồi cuối cùng lại gán NguyenTo:= true gì đó ở trong chương trình. Tất cả chẳng có gì là tự nhiên cả. Nó làm gượng ép suy nghĩ tự nhiên của người học.

Người viết đề xuất: Sử dụng phương pháp phản chứng trong Toán học. Và các bước làm như sau:

Ta sẽ chứng minh mệnh đề: "n là số nguyên tố".

- Bước 1. Giả sử mệnh đề cần chứng minh là sai. Tức giả sử "n không phải là số nguyên tố". Vậy ta gán biến NguyenTo:= false (Bước này gọi là bước phản chứng).
- Bước 2. Ta kiểm tra các số i từ 2 tới n-1 xem có ước nào của n không?
- Bước 3. Nếu n chia hết cho i thì theo định nghĩa số nguyên tố thì n không phải là số nguyên tố. Kết thúc chương trình ngay tại đây.
- Bước 4. Nếu chương trình không kết thúc ở bước 3, mà tiếp tục chuyển sang bước 4 thì có nghĩa là không tìm được ước nào của n từ 2 tới n-1. Vậy suy ra điều giả sử là sai, tức mệnh đề cần chứng minh là đúng. Vậy ta phải gán lại NguyenTo:= true.

Bây giờ, ta sẽ giải thích tại sao lại chỉ đến $\sqrt{n}$.

Thật vậy,

Giả sử: a là một ước của n (a≠1, a≠n). Vậy tồn tại một số b sao cho n=axb. Vậy b cũng là một ước khác của n. Không mất tính tổng quát, giả sử a≤b.

Vậy n=axb≥axa=a^2. Vậy a≤$\sqrt{n}$ (a≠1).

Mặt khác, n=axb≤bxb=b^2. Vậy b≥$\sqrt{n}$ (b≠n).

Như vậy:

(1) Nếu n có một ước lớn hơn $\sqrt{n}$ (ước này khác n) thì n cũng có một ước nhỏ hơn $\sqrt{n}$ (ước này khác 1).

$\Leftrightarrow$

(2) Nếu n không có ước nào nhỏ hơn $\sqrt{n}$ (ước khác 1) thì n cũng không có ước nào lớn hơn $\sqrt{n}$ (ước khác n).

(Theo logic mệnh đề: $A \Rightarrow B \equiv \overline{B} \Rightarrow \overline{A}$)

Trở lại thuật toán kiểm tra số nguyên tố của ta kiểm tra các ước từ 2 tới n-1. Tức là: từ 2→$\sqrt{n}$→n-1. Ta thấy chỉ xuất hiện 2 khả năng:

- Nếu như n chia hết cho số nào từ 2 tới $\sqrt{n}$: Thì theo định nghĩa số nguyên tố thì n không phải là số nguyên tố.
- Nếu như n không chia hết cho số nào từ 2 tới $\sqrt{n}$ thì theo mệnh đề (2) suy ra: n cũng không có ước nào lớn hơn $\sqrt{n}$ (ước này khác n). Tức trong trường hợp này từ 2 tới n-1 không có ước nào của n. Theo định nghĩa số nguyên tố: n là số nguyên tố.

Điều này có nghĩa khi trong thuật toán kiểm tra số nguyên tố ta chỉ cần kiểm tra từ 2 tới $\sqrt{n}$ mà thôi.