Một số bài toán khi nghĩ ra thuật toán nhưng phải nghĩ xem liệu giải thuật đó liệu có đúng và đủ tốt để chạy trong thời gian hợp lí hay không. Một trong những cách chứng minh giải thuật là biểu diễn bài toán bằng 1 hàm số và chứng minh hàm đó luôn tăng hoặc luôn giảm cho đến 1 ngưỡng nào đó.   
Ví dụ: bài GROUPDIV  
 Ta có 1 thuật toán tham như sau:

Chia ngẫu nhiên n con chó vào 2 nhóm ( ví dụ nhóm 1 là những con chó có số hiệu lẻ nhóm 2 số hiệu chẵn, hoặc có thể cả n con chó ở hết trong nhóm 1…)

Trong khi còn có con chó x mà có nhiều hơn 1 con chó bất hòa với x ở chung nhóm với x thì chuyển x qua nhóm khác. Làm như vậy cho đến khi với mọi x <= n, số lượng con chó bất hòa với x mà chung nhóm với x ít hơn 2.

Nhưng liệu giải thuật trên có ổn không, liệu giải thuật trên có bị dính vào vòng lặp vô hạn.

Để chứng minh giải thuật tham này là đúng và chạy được trong thời gian cho phép. Ta gọi hàm F là số cặp chó bất hòa với nhau ở chung nhóm.

Khi ta phải chuyển A từ nhóm x sang nhóm y thì có 2 trường hợp xảy ra:

* Đang có 2 con chó bất hòa với A ở x => khi chuyển A sang y thì có nhiều nhất 1 con chó bất hòa với nó ở y (vì 1 con chó có bất hòa với nhiều nhất 3 con chó khác). Vậy F=F-1.
* Đang có 3 con chó bất hòa với A ở x => không có con chó nào bất hòa với nó ở y. Vậy F=F-3.

Ở trên ta đã chứng minh hàm F luôn giảm và >= 0. Vậy thuật toán trên thực hiện trên đpt O(n)

.

Một cách khác để tính a\*b % c (a, b, c <= 1e18). Dpt O(1)

Code :

Long long mul(long long a, long long b)

{

Long long p = (long double)a\*b/c;

return (a\*b – p\*c + c) % c;

}

Đối với pascal thì phải bỏ thư viện kiểm tra tràn số thì mới có thể thực hiện.

Hướng dẫn bài DIGIT:

Để tìm chữ số k chữ số sau dấu phẩy của : .

* Để tìm chữ số thứ k của

Hướng dẫn REMAINDER

Hướng dẫn SDC : dùng thuật toán bellman ford.  
  
với mỗi rang buộc i , j, c ta nối 1 cung từ i đến j có trong j số là c.

Giả sử   
duyệt n lần, với mỗ lần duyệt m cung nếu v[j] – v[i] > c thì gán .v[i] = v[j] – c.  
trong quá trình làm nếu xuất hiện v[i] > b => không có kết quả thỏa mãn.  
sau n lần duyệt mà vẫn có v[j] – v[i] > c => tồn tại chu trình âm, không có kết quả thỏa mãn.