LỜI GIẢI BÀI THI MÔN CHUYÊN TIN (LẬP TRÌNH)

Thực hiện: Trần Mỹ An, Lê Nhất Duy, Nguyễn Đăng Khoa, Đoàn Thanh Phát, Trương Nguyễn Tấn Phúc

Bài 1: Cấp số cộng – Aripro

Dữ liệu được cho bao gồm:

- Số số hạng n
- Số nguyên a_1
- Số nguyên a_2

➔Công sai $d = a_2 - a_1$

Từ đó, ta dễ dàng tìm được:

- Số hạng thứ n của cấp số cộng theo công thức:

$$a_n = a_1 + (n-1) \times d$$

- Tổng n số hạng đầu tiên của cấp số cộng theo công thức:

$$S_n = \frac{n \times [2 \times a_1 + (n-1) \times d]}{2}$$

Độ phức tạp: O(1)

Bài 2: Trà sữa Bmilk – Bmilk

Ta sẽ kiểm tra với mỗi k khách hàng thì số tiền thu được có lớn hơn hoặc bằng p hay không và cập nhật số tiền trong ống heo.

Lưu ý:

- Số tiền có thể rất lớn nên phải sử dụng kiểu dữ liệu phù hợp;
- Giải quyết trường hợp nhóm khách hàng cuối có thể không đủ k người.

Độ phức tạp: O(n)

Bài 3: Bài tập về nhà – Homework

Một số tự nhiên N có 3 ước khi và chỉ khi số đó là số chính phương và $\sqrt{N}$ là một số nguyên tố

Chứng minh:

(1) N là bình phương số nguyên tố $=>$ N có 3 ước nguyên dương. Thật vậy , $N = p^2$ (p là số nguyên tố) có đúng 3 ước nguyên dương là $1, p, p^2$.

(2) N có 3 ước nguyên dương $=>$ N là bình phương của một số nguyên tố. Giả sử N không là bình phương của một số nguyên tố. Khi đó $N = p1^{k1} * p2^{k2} * .... * pm^{km}$, trong đó pi là các số nguyên tố ($m \geq 2$ và $Ki \geq 1$). Nên số ước của N là $\sum_{i=1}^{m}(Ki + 1) \geq 4$ (mẫu thuẫn). Qua phản chứng ta có điều phải chứng minh.

Từ (1) và (2) $=>$ Một số tự nhiên N có 3 ước khi và chỉ khi số đó là số chính phương và $\sqrt{N}$ là một số nguyên tố.

Lời giải: Sau khi đọc vào mảng a ta tìm max của mảng sao đó sàng nguyên tố đến $10 * [\sqrt{max}]$ tại sao lại phải là $10 * [\sqrt{max}]$? . Bởi vì có 1 số trường hợp khi rút căn dẫn đến sai số nên ta phải $* 10$ để tránh việc sai số ở đây . Sau đó với mỗi phần từ $a[i]$ ta chỉ cần kiểm tra nếu $[\sqrt{a[i]}]$ phải là 1 số nguyên tố và $[\sqrt{a[i]}] * [\sqrt{a[i]}] = a[i]$ thì ta tăng biến res lên 1. Kết quả cuối cùng là res. Để tăng tốc độ chạy ta có thể sử dụng các loại sàng nguyên tố có tốc độ tốt như : sàng Eratosthenes $O(Nlog(log(N)))$.

Độ phức tạp: $O(NlogN)$

Bài 4. Tìm xâu Palindrome – Palin

Đầu tiên, ta có thể nhận ra thuật toán $O(n^2)$: tại mỗi vị trí của xâu *st*, ta kiểm tra xem tại vị trí đó có phải là bắt đầu của một xâu palindrome có độ dài k hay không, nếu có thì ta tăng biến đếm cnt. Cho đến khi biến đếm cnt = i, ta in ra kết quả và kết thúc.

Để kiểm tra 1 xâu có phải là palindrome hay không, ta chỉ cần chạy 2 con trỏ: con trỏ bên trái ở đầu xâu và con trỏ bên phải ở cuối xâu. Nếu 2 ký tự ở vị trí

hai con trỏ giống nhau thì ta tăng trỏ trái và giảm trỏ phải và lặp lại cho đến giữa xâu. Khi đó, ta kết luận xâu đó là xâu palindrome.

Do độ dài xâu *st* lên đến 10^5 nên để đạt full test bài này, ta có thể cải tiến thuật toán ở trên bằng cách tối ưu phép kiểm tra xâu palindrome ở trên từ O(n) xuống O(1).

Cụ thể hơn, việc kiểm tra 2 xâu con bằng nhau hay không sẽ phải tốn mất nhiều nhất O(k/2). Để tối ưu việc kiểm tra 2 xâu trong O(1), ta cần dùng kỹ thuật Hàm băm xâu (String Hashing).

Ta định nghĩa 1 hàm băm của xâu st có độ dài m là:

$$hash(st) = st[0] + st[1] * p + st[2] * p^2 + \cdots + st[n-1] * p^{n-1} \ mod \ m$$

$$= \sum_{i=0}^{n-1} st[i] * p^i \ mod \ m$$

Vì xâu st của ta chỉ bao gồm ký tự chữ cái Latin thường, ta sẽ dùng $p = 31$ và modulo $m = 10^9 + 9$

Để lấy mã Hash của 1 xâu con $s[l \ldots r]$, ta định nghĩa hàm $GetHash(l, r)$ là

$$(Hash[r] - \ Hash[l-1] * p^{r-l+1} + m * m) \ mod \ m$$

Ví dụ: để kiểm tra xâu "abcba" là palindrome hay không, vì ta cần phải so sánh 2 xâu con "ab" và xâu đảo ngược "ba" nên ta cần tìm hàm băm xuôi và ngược của cả xâu st.

Gọi 2 mảng $Hash\_L[0 \ldots n]$ và $Hash\_R[0 \ldots n]$ là hàm băm xuôi và ngược của xâu st và 2 function $GetHashL(l, r)$ và $GetHashR(l, r)$ để lấy mã Hash trong đoạn $[l, r]$ của 2 hàm băm trên. Để kiểm tra tại ví trí thứ j của xâu st có phải là bắt đầu của một xâu palindrome có độ dài k hay không, ta xét 2 trường hợp:

+ k là số chẵn: kiểm tra

$$GetHashL\left(j \ , j + \frac{k}{2} - 1\right) = GetHashR(j + \frac{k}{2} \ , j + k - 1)$$

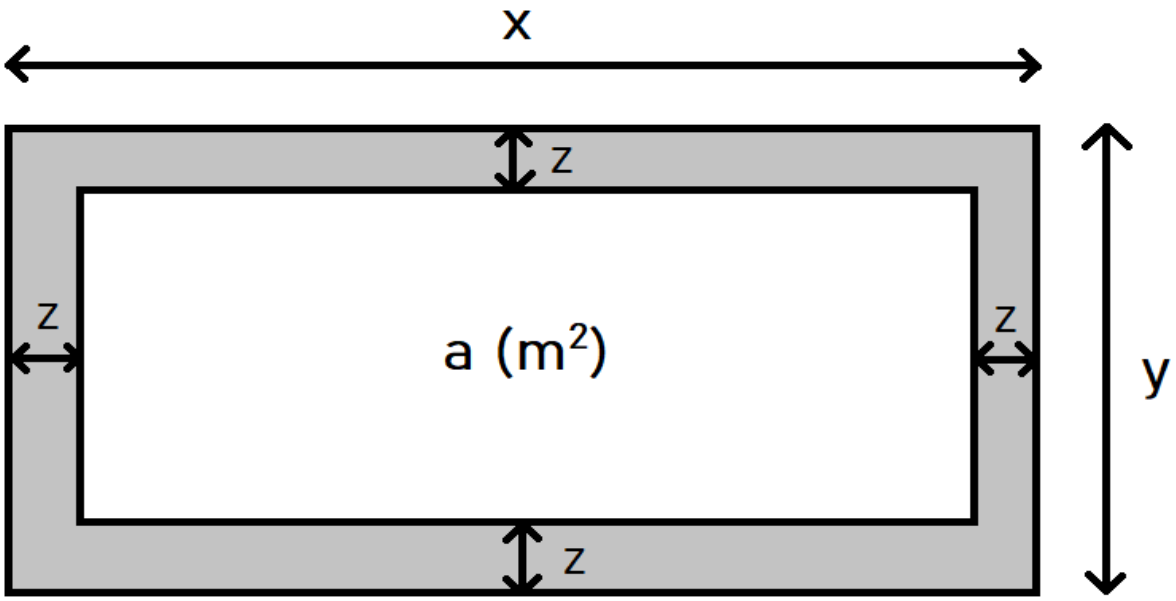
+ k là số lẻ: kiểm tra

$$GetHashL\left(j\ , j+\frac{k}{2}-1\right) = GetHashR(j+\frac{k}{2}+1\ , j+k-1)$$

(ta lưu ý C++ làm tròn xuống trong phép chia số nguyên)

Tiền xử lý hàm băm tốn O(2*n) + xét mỗi phần tử của xâu st tốn O(n) * O(1) cho phép kiểm tra palindrome -> Độ phức tạp tổng : O(n)

Bài 5: Lối đi quanh công viên – Park



Với bài này ta dễ dàng nhận thấy công việc cần phải thực hiện là tìm z khi đã biết được các thông số x, y, a. Phân tích ta thấy:

Phần diện tích vườn trồng hoa là một hình chữ nhật có chiều dài là tổng chiều dài của cả công viên trừ cho 2 lần độ dài z (như hình vẽ) là $x - 2z$ (m) và có chiều rộng là chiều rộng cả công viên trừ cho 2 lần độ dài z (như hình vẽ) là $y - 2z$ (m). Như vậy diện tích phần trồng hoa:

$$a = (x - 2z) \times (y - 2z)$$

$$\Leftrightarrow \quad a = xy - x(2z) - (2z)y + 4z^2$$

$$\Leftrightarrow \quad a = xy - 2z(x+y) + 4z^2$$

$$\Leftrightarrow \quad 4z^2 - 2z(x+y) + (xy - a) = 0$$

Với x, y, a đề cho trước, dễ dàng ta giải tìm z bằng cách giải phương trình bậc 2.

Tổng kết: 5 bài thi có độ khó không quá cao, tuy nhiên cũng không dễ để có thể đạt được số điểm tối đa. Đây là năm đầu tiên trường THPT Chuyên Tiền Giang thực hiện tuyển sinh chuyên tin theo hình thức lập trình nên sẽ làm cho các thí sinh còn mới lạ chưa quen với cấu trúc đề. Dù sao thì vẫn chúc mừng tất cả các bạn đã hoàn thành xuất sắc kỳ thi tuyển sinh 10! Chúc các bạn đạt được kết quả như mong muốn nhé!

Bài giải được thực hiện bởi ban chuyên môn DevGang – câu lạc bộ lập trình Chuyên Tiền Giang. Bài giải chỉ mang tính chất tham khảo vì mỗi bài sẽ có nhiều cách làm.