HƯỚNG DẪN GIẢI CODE CHALLENGE 1
KỲ THI CODE TOUR 2022

BÀI A: NHỮNG VIÊN BI

Solution:

C++	https://ideone.com/l8guZe

Tóm tắt đề:

Cho mảng A gồm N phần tử, và cho một số nguyên dương X. Độ đẹp của một đoạn con từ $[l,r]$ là:

$$F(l,r) = \left(\sum_{l \le i \le r} |A_{i+1} - A_i|\right) \cdot (r - l)$$

Đếm xem có bao nhiêu đoạn $[l,r]$ mà $(l < r)$ thoả mãn $F(l,r) \ge X$

Input:

Dòng đầu tiên là 2 số nguyên dương N và X.

Dòng tiếp theo chứa N số nguyên dương A_i.

Điều kiện:

- $2 \le N \le 10^5$
- $1 \le X \le 10^{15}$
- $1 \le A_i \le 10^7$

Output:

Một dòng duy nhất chứa số lượng đoạn $[l,r]$ thoả mãn.

Ví dụ:

6 5 1 5 2 3 1 6	11

Hướng dẫn giải:

Đầu tiên ta xây dựng mảng B gồm $N-1$ phần tử với $B_i = |A_{i+1} - A_i|$. Khi đó bài toán trở thành đếm số bộ (l, r) mà $(l \leq r)$ thoả mãn.

$$\left(\sum_{l \leq i \leq r} B_i\right) \cdot (r - l + 1) \geq X$$

Tạm gọi công thức ở trên là $G(l, r)$. Khi đó ta nhận thấy rằng nếu $G(l, r) \geq X$ thì $G(l-1, r) \geq X$. Do đó ta có thể có hướng giải quyết là với mỗi $r = [1..N-1]$ ta tìm kiếm nhị phân vị trí l gần với r nhất thoả mãn $G(l, r) \geq X$. Khi đó với mọi $i \in [1..l]$ thì đều thoả mãn $G(i, r) \geq X$ nên ta sẽ cộng vào kết quả một lượng là l tương ứng.

Ngoài ra ta có thể sử dụng kĩ thuật 2 con trỏ thay vì tìm kiếm nhị phân để có thể tối ưu hoá thuật toán.

Độ phức tạp:

- **O(N)** khi kĩ thuật 2 con trỏ.
- **O(NlogN)** khi sử dụng tìm kiếm nhị phân.

BÀI B: VÍ ĐIỆN TỬ ZALOPAY

Solution:

C++	https://ideone.com/kDicr1
Python	https://ideone.com/rr9a1p

Tóm tắt đề:

Cho một ma trận $N \times N$ như hình sau:

1	2	3	...	N
$N+1$	$N+2$	$N+3$	...	$2N$
$2N+1$	$2N+2$	$2N+3$	...	$3N$
...	...	...	...	...
$(N-1)N+1$	$(N-1)N+2$	$(N-1)N+3$	...	N^2

Với $1 \leq N \leq 10^{18}$.

Chọn ra N ô trong ma trận sao cho hai ô được chọn bất kỳ không được trùng hàng hoặc cột.

Tìm tổng N ô được chọn lớn nhất có thể, do tổng có thể rất lớn nên chia lấy dư cho 2027.

Input:

Chỉ duy nhất một dòng chứa một số N $(1 \leq N \leq 10^{18})$.

Output:

Tổng lớn nhất có thể chọn ra, chia lấy dư cho 2017.

Ví dụ:

566	1429
5	65

Hướng dẫn giải:

Trước hết ta sẽ chứng minh rằng tổng của N ô được chọn ra bất kì đều như nhau, từ đó tìm ra tổng đó và tính đáp án của mình cần:

Chứng minh:

Nhận xét thấy từ hình vẽ đề cho

1	2	3	...	N
$N+1$	$N+2$	$N+3$	...	$2N$
$2N+1$	$2N+2$	$2N+3$	...	$3N$
...	...	...	...	...
$(N-1)N+1$	$(N-1)N+2$	$(N-1)N+3$	...	N^2

- Đánh số từng hàng từ 1 đến N và từng cột từ 1 đến N, như vậy giá trị của một ô ở hàng I, cột j sẽ là $A_{i,j} = (i-1) \times N + j$.
- Vậy N số ta chọn từ ma trận sẽ là một dãy $A_{1,\alpha_1}, A_{2,\alpha_2}, A_{3,\alpha_3}, \dots A_{n,\alpha_n}$ với dãy $\{\alpha_i\}$ là hoán vị của các số từ 1 đến N.
- Ta suy ra được:

$$\sum_{i=1}^{N} A_{i,\alpha_i} = \sum_{i=1}^{N} (i-1) \times N + \alpha_i = N \times \sum_{i=1}^{N} (i-1) + \sum_{i=1}^{N} \alpha_i$$

$$= N \times \frac{N \times (N-1)}{2} + \frac{N \times (N+1)}{2}$$

- Vậy với một N cho trước, ta có thể tính được tổng chính là $\frac{N \times (N^2+1)}{2}$
- Lúc này, bài toán chỉ còn là tính $\frac{N \times (N^2+1)}{2} \ mod\ 2027$.

Cách 1:

- Nhận xét thấy là $\gcd(2, 2027) = 1$ nên ta có thể tính nghịch đảo modulo 2027 của 2 (là một số nguyên x sao cho $2 \times x \equiv 1\ mod\ 2027$). Khi đó ta có thể tính $2^{-1} \equiv x\ mod\ 2027$ và tính tổng khi đó sẽ là $N \times (N^2 + 1) \times x\ mod\ 2027$.

Cách 2:

- Hoặc ta sẽ không tính nghịch đảo Modulo 2027 mà ta sẽ khử đi số 2 ở dưới mẫu bằng cách xét TH chẵn và lẻ.
    - Nếu N lẻ, khi đó ta sẽ có $N = 2k + 1$ với k là một số nguyên. Thế vào trong công thức tính tổng ta có

      $$\frac{N\times((2k+1)^2+1)}{2} = N \times \frac{4k^2+4k+2}{2} = N \times (2k^2 + 2k + 1)\ mod\ 2027.$$

    - Nếu N chẵn, khi đó ta sẽ có $N = 2k$ với k là một số nguyên và công thức tổng của ta sẽ thành $k \times (N^2 + 1)\ mod\ 2027$

Cách 3:

- Ngoài ra thì chúng ta thấy rằng con số chia lấy dư 2027 tương đối nhỏ, vậy chúng ta thử suy ngược bài toán xem sao.
    - Thay vì chúng ta tính trực tiếp $N \times \frac{N^2+1}{2}\ mod\ 2027$ thì chúng ta thử hỏi liệu một số m có phải là đáp án?
    - Nếu m chính là đáp án thì chúng phải thỏa $N \times \frac{N^2+1}{2} \equiv m\ (mod\ 2027)$. Do $\gcd(2,2027) = 1$ nên ta có thể nhân 2 cho hai vế thành $N \times (N^2 + 1) \equiv 2m\ (mod\ 2027)$. Như vậy ta có thể giải quyết bài toán bằng cách cho m chạy từ 0 đến 2026. Nếu m nào thỏa điều kiện trên sẽ in ra đáp án là m đó.

Độ phức tạp: **O(1)** do chỉ sử dụng công thức để tính ra đáp án.

BÀI C: UPIN VÀ IPIN

Solution:

Python	https://ideone.com/mDw0BL

Tóm tắt đề:

Ipin tham gia vào trò chơi của Upin với luật chơi như sau:

- Có n lượt chơi. Hộp chứa bi ban đầu rỗng
- Ở mỗi lượt Ipin có 2 lựa chọn:
  - Thêm bi vào một chiếc hộp (ban đầu rỗng) với điều kiện lần thêm đầu tiên là 1 bi. Mỗi lần thêm tiếp theo mỗi lần phải thêm nhiều hơn c bi so với lần thêm trước.
  - Lấy đi 1 bi từ hộp.
- Lượt chơi đầu luôn là thêm bi

Ipint sẽ phải thực hiện các thao tác sao cho trong hộp còn chứa đúng k bi sau n lượt chơi. Tìm số bi Ipin nhận đc sau n lượt chơi.

Input:

Dòng duy nhất chứa 3 số nguyên n, k, c lần lượt là tổng số lượt chơi, số bi còn lại trong hộp và số bi tăng lên sau mỗi lần thêm.

Điều kiện:

- $1 \le n \le 10^9$
- $0 \le k \le 10^9$
- $1 \le c \le 100$

Output:

Số bi mà Ipin nhận được như là phần thưởng của mình

Chú ý: Đề đảm bảo kết quả là duy nhất và luôn tồn tại kết quả.

Ví dụ:

9 11 1	4

Giải thích ví dụ:

Trò chơi diễn ra trong 9 lượt:

- Lượt 1: Thêm 1 bi vào hộp

- Lượt 2: Thêm $1 + 1 = 2$ bi vào hộp. Hộp đang có 3 bi
- Lượt 3: Thêm $1 + 1 + 1 = 3$ bi vào hộp. Hộp đang có 6 bi
- Lượt 4: Lấy 1 bi từ hộp. Hộp còn 5 bi
- Lượt 5: Lấy 1 bi từ hộp. Hộp còn 4 bi
- Lượt 6: Thêm $1 + 1 + 1 + 1 = 4$ bi vào hộp. Hộp còn 8 bi
- Lượt 7: Lấy 1 bi từ hộp. Hộp còn 7 bi
- Lượt 8: Thêm $1 + 1 + 1 + 1 + 1 = 5$ bi vào hộp. Hộp còn 12 bi
- Lượt 9: Lấy 1 bi từ hộp. Hộp còn 11 bi

Kết luận: Có 4 lượt lấy bi, Ipin nhận được 4 bi.

Hướng dẫn giải:

Nhận xét:

Trong trường hợp lấy bi hợp lệ (hộp không rỗng), nếu đổi thứ tự các lượt lấy bi và thêm bi thì số bi còn lại trong hộp cuối cùng vẫn không đổi.

Gọi u là số lần thêm bi vào hộp. Suy ra $n - u$ là số lượt lấy bi ra cũng là đáp án cuối cùng của đề bài.

Với u lượt thêm bi, tổng số bi thêm vào hộp là:

$add = 1 + (1 + c) + (1 + 2c) + (1 + 3c) + .. + (1 + (u - 1)c)$

$add = u + c(1 + 2 + 3 + .. + (u - 1))$

$$add = u + \frac{cu(u-1)}{2}$$

Với $n - u$ lượt lấy bi, tổng số bi lấy ra khỏi hộp là:

$$take = (n - u) * 1 = n - u$$

Ta có phương trình như sau: $add - take = k$

Cách 1: Sử dụng thuần kiến thức toán để giải phương trình bậc 2 trên. Tuy nhiên chú ý về việc làm tròn số khi lấy căn có.

Cách 2: Để tránh việc sai số khi lấy căn và làm tròn, ta có thể dùng binary search từ $1 \rightarrow n$ để tìm ra đáp án phù hợp từ phương trình trên.

Chú ý: $n - u$ là kết quả cuối cùng của bài toán.

Độ phức tạp:

Time Complexity: $\boldsymbol{O(log(n))}$ cho cách 2 và $\boldsymbol{O(1)}$ cho cách 1.

Space Complexity: $\boldsymbol{O(1)}$

BÀI D: THE MATRIX

Solution:

C++	https://ideone.com/PlZgHh

Tóm tắt đề:

Có một ma trận vuông A kích thước $N \times N$, ban đầu gồm toàn số 0. Bạn muốn biến nó thành một ma trận B cũng có kích thước $N \times N$. Bạn được quyền thực hiện phép biến đổi sau: - Chọn một hoán vị P có kích thước N. - Chọn một số nguyên **dương** X bất kì. - Với mỗi i $(1 \leq i \leq N)$, cộng X vào A_{i,P_i}.

Một hoán vị có kích thước N được định nghĩa là một dãy số nguyên dương, trong đó các phần tử đều mang giá trị từ 1 đến N, và không có hai phần tử nào có giá trị bằng nhau.

Bạn được cho ma trận B. Bạn được quyền thực hiện nhiều nhất 2500 lần biến đổi. Hãy cho biết bạn có thể biến đổi ma trận A thành ma trận B hay không? Nếu được, bạn hãy cho biết một cách để thực hiện việc đó.

Input:

Dòng đầu tiên của input chứa số nguyên T, là số lượng test.
Dòng đầu tiên của input chứa số nguyên N, là kích thước của ma trận.
N dòng tiếp theo, dòng thứ i chứa N số nguyên, là các phần tử của dòng thứ i trong ma trận B theo thứ tự.

Output:

Với mỗi test: - Nếu không có cách nào để tạo ra đúng ma trận, hãy in ra duy nhất một số -1.

Nếu có đáp án, trước hết hãy in ra một số nguyên K $(0 \leq K \leq 2500)$. K dòng tiếp theo, in ra $N + 1$ số nguyên dương, mô tả các phép biến đổi theo thứ tự như sau:

- Số nguyên dương đầu tiên là giá trị X của phép biến đổi. $(1 \leq X \leq 10^9)$.
- N số nguyên dương tiếp theo mô tả hoán vị P theo thứ tự.

Nếu có nhiều đáp án, bạn có thể đưa ra đáp án bất kì.

Ví dụ:

3 4 1 1 1 1 0 2 0 2 2 0 1 1 1 1 2 0 3 1 2 3 4 5 6	4 1 2 4 3 1 1 3 4 1 2 1 4 2 1 3 1 1 2 4 3 -1 1 314159 1

7 8 9 1 314159	

Giới hạn:

- $1 \leq T \leq 5$
- $1 \leq N \leq 50$
- $0 \leq B_{i,j} \leq 10^9$

Hướng dẫn giải:

Điều kiện cần để có đáp án, là tổng của tất cả các hàng bằng nhau, và tổng tất cả các cột bằng nhau.

Đây cũng là điều kiện đủ.

Để giải bài này, mỗi lượt đi, ta sẽ tạo một đồ thị 2 phía gồm $2N$ đỉnh. Sẽ có một cạnh nối i trái với j phải nếu như $B_{i,j} > A_{i,j}$. Một bộ ghép đầy đủ của đồ thị này chính là một nước đi ta cần.

Có thể chứng minh bằng định lý Hall rằng luôn có một bộ ghép đầy đủ nếu điều kiện này thỏa mãn. Chứng minh như sau:

Ta gọi một ma trận "cân bằng" là ma trận thỏa mãn tổng mỗi hàng bằng tổng mỗi cột, và bằng nhau. Ta khởi đầu với ma trận A gồm toàn số 0. Mỗi bước, ta cộng vào A một ma trận cân bằng. Dễ thấy rằng tổng của tất cả các ma trận cân bằng phải là một ma trận cân bằng. Như vậy, nếu ma trận B không cân bằng, thì luôn không có đáp án.

Ở trường hợp còn lại, ma trận B là cân bằng. Ta sẽ chứng minh luôn tồn tại một cặp ghép thỏa mãn.

Dựng một đồ thị hai phía có $2N$ đỉnh. Từ đỉnh i trái đến đỉnh j phải sẽ có $B_{i,j} - A_{i,j}$ cạnh nối (như vậy đây là đa đồ thị). Do ma trận $B - A$ cân bằng, ta có bậc của mỗi đỉnh là bằng nhau, tạm gọi là bằng X.

Xét bất cứ một tập hợp con gồm K đỉnh bất kì bên trái. Tập hợp này kề với đúng $K * X$ đỉnh. Như vậy bên phải cũng cần ít nhất K đỉnh mới có thể kề được với đủ $K * X$ cạnh. Như vậy điều kiện của định lý Hall được thỏa mãn, và đồ thị này luôn có cặp ghép đầy đủ (trừ khi $X = 0$, trong trường hợp bài toán đã được giải xong).

---------- HẾT ----------