08/01/2022

BN 2022 Day 01.

A. NCYRCLES

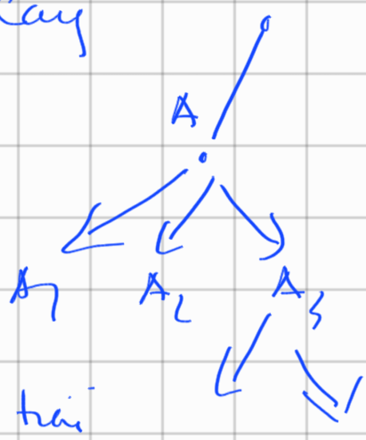
Cho n đường tròn có tâm trên trục hoành, không cắt nhau. Hỏi rằng n hình tròn này chia mặt phẳng thành bao nhiêu phần?

Định nghĩa quan hệ nằm trong có nghĩa đường tròn A nằm trong đường tròn B

+) Nằm trong trực tiếp (=> A nằm trong B

∄ C: A nằm trong C và C nằm trong B.

=> Tập các đường tròn có cấu trúc dạng hình cây



Mỗi đường tròn sẽ làm số phần mặt phẳng tăng thêm 1 nếu như $\sum d < d$(Hình Tròn)

Nếu $\sum d(\text{con}) = d(\text{đường tròn})$ => +2.

Cách 1: Dựng trường hình cây (Sweep Line từ trái sang phải)
- Bắt đầu đường tròn $(x-r)$
- Kết thúc đường tròn $(x+r)$

Với mỗi đỉnh u thì $v_1, v_2, \ldots, v_k$ là con của nó

KQ: $d(u) = \sum d(v_i)$ +2 | Cuối cùng +1.

$< \sum d(v_i)$ +1. | (Toàn mặt phẳng)

Cách 2: DFS trên cây (=> quét từ trái sang phải)
(Sử dụng 1 stack lưu dấu (, đường kính

B. Hash

Xây dựng $f(\emptyset) = 0$

$f(word + c) = ((f(word) \times 33) \wedge ord(c)) \% MOD$

Yêu Cầu: Đếm xem có bao nhiêu xâu độ dài N mà giá trị hàm băm bằng K.

Sub1 $N \le 5$. => Thử tất cả xâu độ dài N. Có $26^N \le 26^5$

Với mỗi xâu ta tính hàm băm => đếm số lượng $= K$.

Sub2: Chia xâu $S = s_1 s_2 \dots s_N$ thành 2 phần

1) + Phần tiền tố $s_1 s_2 \dots s_{N/2}$. (A)

2) + Phần hậu tố $s_{N/2+1}, \dots, s_N$ (B)

Với (1): Xây dựng tất cả các tiền tố với mỗi tiền tố ta tính hàm băm của nó => đếm số lượng tiền tố có giá trị băm là $x \in [0, 2^M - 1]$. => Lưu vào mảng $a[x]$

(2) Duyệt qua tất cả các hậu tố

Với 1 hậu tố $s_1 s_2 \dots s_q$

$$f\left(\underbrace{\dots\dots}_{f(-)=u} s_1 s_2 \dots s_q\right) = K.$$

Giải bài toán ngược: Biết hậu tố, biết giá trị băm K => Tìm giá trị băm u của tiền tố => Ds += $a[u]$

=> Cần giải bài toán ngược:

Biết $f(S) = x$ $\quad f(S + c) = ((x * 33) \,\hat{}\, c) \,\%\, MOD = y$

BT Ngược: Biết y Hãy tìm x.

NX1 MOD = 2^M $\Rightarrow$ p % MOD $\Leftrightarrow$ giữ lại M bit đầu tiên của p (bỏ qua các bit cuối)

$\Rightarrow$ % MOD đơn giản là lấy M bit đầu tiên (0 $\rightarrow$ M-1)

Hệ quả $(a \wedge b) \% MOD = (a \% MOD) \wedge (b \% MOD)$

Áp dụng: $((x * 33) \wedge c) \% MOD = ((x*33) \% MOD) \wedge c = y$

$\Rightarrow (x * 33) \% MOD = y \wedge c$

(Chú ý là tất cả các giá trị chỉ giữ lại M bit đầu)

Do $\gcd(33, 2^M) = 1 \Rightarrow$ Có thể tìm giá trị nghịch đảo INV của 33 (Euclid mở rộng)

$$\Rightarrow \boxed{x = (y \wedge c) * INV}$$

$a \wedge a = 0$

$a \wedge 0 = a$

$MOD = 2^3 = 8$

13 % 8 = 5

1 101

Nghịch đảo modun của a theo đồng dư P được định nghĩa là số x: $a.x \equiv 1 \pmod{P}$ ($x = a^{-1}$)

(*) Thuật toán Euclid mở rộng.

Định lý: Đặt $d = \gcd(a, b) \Rightarrow \exists\, x, y \in \mathbb{Z}$

sao cho $d = ax + by$.

Thuật Toán Euclid mở rộng cho phép tìm d, x, y.

$$d = \gcd(a, b) = \gcd(b, a \% b).$$

$$\Leftrightarrow d = ax + by = b.x' + (a \% b) y'$$

$$= bx' + [a - (a/b).b] y'$$

$$= a.y' + b.[x' - (a/b).y']$$

Có thể lấy $\begin{cases} x = y' \\ y = x' - (a/b) y' \end{cases}$

```
void E_gcd(int a, int b, int &d, int &x, int &y) {
    if (b == 0) { x = 1, d = a, y = 0; ret }
    int x1, y1;
    E_gcd(b, a%b, d, x1, y1);
    x = y1;
    y = x1 - (a/b) * y1;
}
```

Quay Lại: Tìm x: $ax \equiv 1 \pmod{P}$ đk $\gcd(a, P) = 1$

Do $\gcd(a, P) = 1 \Rightarrow \exists u, v \in \mathbb{Z}$

Sao cho $1 = a.u + P.v$

$$1 = a.u \pmod{P}$$

$\Rightarrow$ Lấy $x = (u \% P + P) \% P$ là nghịch đảo của a.

Truy vết ngược lấy các hoán vị

Khi có 1 hoán vị $s_1 s_2 \ldots s_q$.

```
void Truy() {
    u = K;
    for (i = q; i >= 1; --i) {
        u = (u ^ (s_q - 'a' + 1)) * INV;
    }
    Ds += a[u];
}
```

C. TIRES.

Cho $n+1$ răng dưới chia thẳng $a_1, a_2, \ldots, a_{n+1}$.

Đếm xem cần bao nhiêu đg cắt khác nhau. Mỗi đg cắt nằm trên 1 đĩa cho.

Xét một răng chia thẳng P phần

=> Đĩ các quan tâm

$1/P,\ 2/P,\ 3/P,\ \ldots,\ (P-1)/P$.

Bài toán quy về

Duyệt qua các răng từ $1 \to n+1$.

$P = a[i]$

Ds += Các phần tử chưa xuất hiện trong tập $\{1/P, 2/P, \ldots, (P-1)/P\}$.

Làm thế nào để có thể đếm mà không cần liệt kê từng phần tử

VD: $P = 10$

$1/10,\ 2/10,\ 3/10,\ 4/10,\ 5/10,\ 6/10,\ 7/10,\ 8/10,\ 9/10$

$1/10 \quad 1/5 \quad 3/10 \quad 2/5 \quad 1/2 \quad 3/5 \quad 7/10 \quad 4/5 \quad 9/10$

Tử số ngtố cùng nhau với 10

5

2.

Kết luận: Giả sử d là ước của P. Khi đó nếu d chưa xuất hiện (là ước của một số trước đó)

=> Ds += số lượng các số $< d$ ngtố cùng nhau với d.

=> Ds += $\phi(d)$ trong đó $\phi(d)$ là phi hàm Euler.

Lưu ý: _ Duyệt qua các ước > 1 của P cần $O(\sqrt{P})$

_ Để nhớ ước nào đã xuất hiện dùng 1 mảng

```
int nho[maxA];
for(i=1; i<=n+1; ++i) { P = a[i];
    vector<int> d; d.pb(P);
    for(u=2; u*u <= P; ++u) if (P%u==0)
    { d.pb(u);
      if (u*u < P) d.pb(P/u);
    }
    for(u : d) if (nho[u]==0)
    { Ds += φ(d); nho[u]=1 };
}
```

D. Stdorm.

<u>NX</u>: Vì mỗi ngôi nhà ta chỉ quan tâm đến số lượng sinh viên có trong ngôi nhà đó (Thứ tự không quan trọng)

<u>BT1</u>: Giả sử một ngôi nhà có S sinh viên tới ở và được phép được tối đa p lần.
Hỏi rằng lượng tiếng ồn tối thiểu = ?

$$x_1 + x_2 + \dots + x_{p+1=k} = S$$

Tổng lượng ồn $= (1+2+\dots+x_1) + (1+2+\dots+x_2) + \dots + (1+2+\dots+x_k)$

$$= \frac{1}{2}\left[x_1(x_1+1) + x_2(x_2+1) + \dots + x_k(x_k+1)\right]$$

$$= \frac{1}{2}\left[(x_1^2 + x_2^2 + \dots + x_k^2) + (x_1 + x_2 + \dots + x_k)\right]$$

$$= \frac{1}{2}\left[(x_1^2 + x_2^2 + \dots + x_k^2) + S\right]$$

Tổng tiếng ồn bé nhất khi $x_1^2 + x_2^2 + \dots + x_k^2 \to \min$

<u>Kquả</u>. Giá trị bé nhất đạt được khi $x_1 = x_2 = \dots = x_k$.

Khi các x_i nguyên thì giá trị bé nhất đạt được khi các x_i chênh nhau ≤ 1.

```
int calc(S, P) {
    k = P + 1;
    T = S / k;
```

```
f(T) = T(T+1)/2;
```

$D = S \% K$

} return $D * f(T+1) + (K-D) * f(T)$;

Giả sử có M ngôi nhà => Sử dụng Dynamic.

Đặt $dp[i,j]$ = giá trị nhỏ nhất khi xét i ngôi nhà đầu tiên và chia không quá j lần

$$dp[i,j] = \min_{0 \le u \le j} \{ dp[i-1, j-u] + calc(a[i], u) \}$$

$O(mk^2)$

Đáp số $dp[m,k]$.

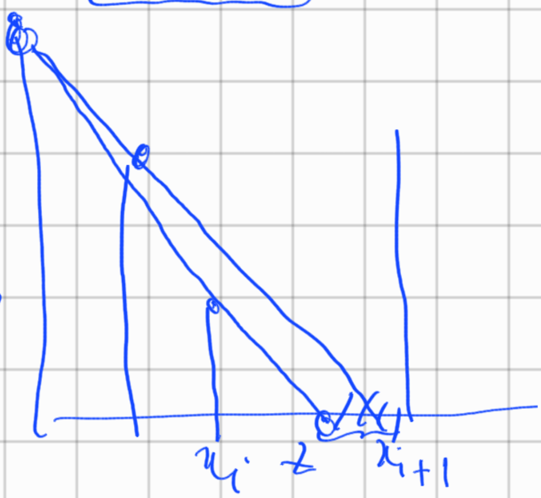
E. WIFI   Tư tưởng chia thành 2 lượt

Lượt 1: Giả sử các trạm phát WIFI chỉ phát sang bên phải

Xét đoạn $[x_i, x_{i+1}]$ giữa ngôi nhà thứ i, i+1.

=> Các trạm có thể phủ sóng đoạn này phải nằm ở các ngôi nhà bên trái đoạn $[x_i, x_{i+1}]$

Với mỗi vị trí bên trái gọi $[\underbrace{z, x_{i+1}}]$ là đoạn nó phủ.



Đặt z_L là $\min\{z\}$

=> Các trạm bên trái phủ $[x_i, x_{i+1}]$ bằng đoạn $[z_L, x_{i+1}]$

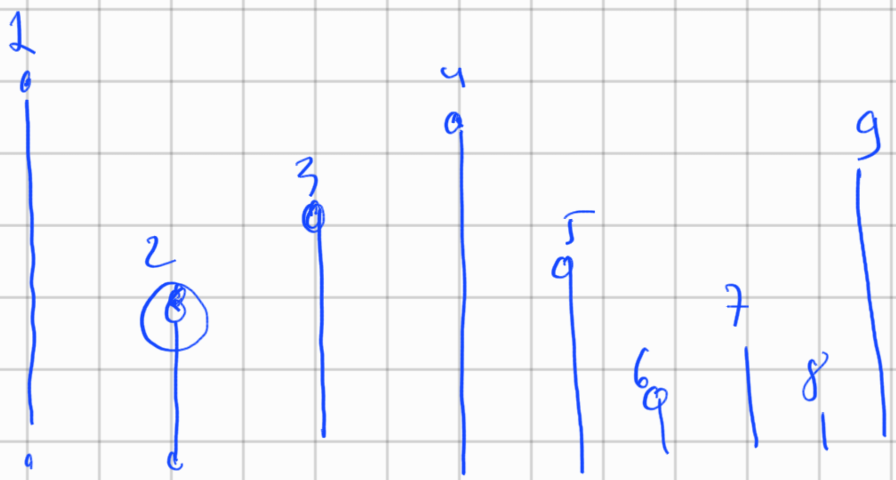
Loại 2: Làm tương tự với các tham wifi bên phải

=> với 1 đoạn $[x_i, x_{i+1}]$ có đoạn phủ $[x_i, z_R]$

Tổng kết $\quad D_s = \sum_i \left(x_{i+1} - x_i - \max(0, z_L - z_R)\right)$

Cách giải pháp hiệu quả (xét Loại 1)

Lưu trữ các wifi



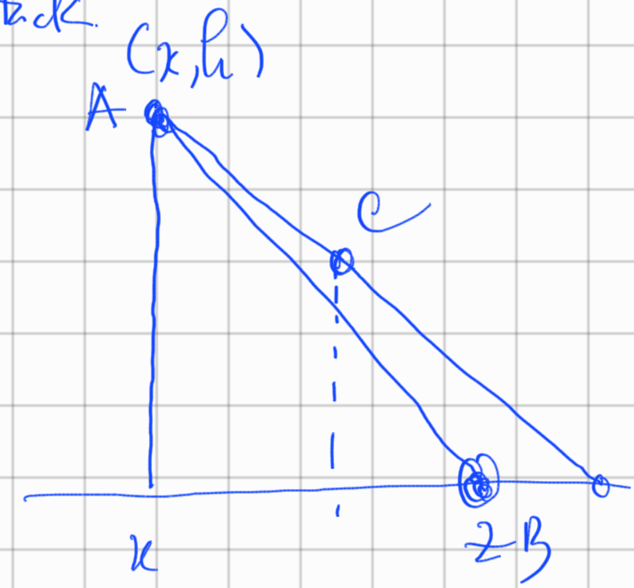
1 ~~2~~ ~~3~~ 4 ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9

Lưu trữ các wifi bên trái trong 1 cấu trúc giống stack

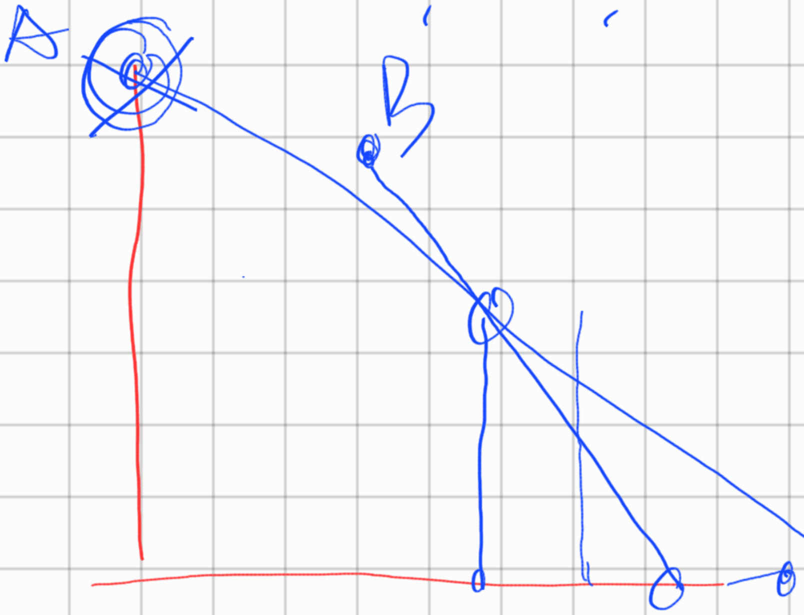
=> Trong 1 stack các wifi luôn lưu trữ độ cao giảm dần từ đỉnh stack đến đáy stack.

(*) Thông tin về 1 wifi

```
struct wifi{
    double x, h, z;
}
```



Khi mà người nhà xuất hiện thì cần dừng lại giá trị z của vị trí



Nếu A có z > z của B.

Thì có thể bỏ A đi

```
vitri st[maxn];
sn = 0;
for (i = 1; i <= n; ++i) { while (sn>0 && s[sn].h <= hi) --sn;
    while (sn > 0) { upd (s[sn], (xi, hi));
        if (sn > 1) upd((s[sn-1], (xi, hi));
        if (s[sn].z <= s[sn-1].z) break;
    --sn;
    }
    if (b[i] == 1) s[++sn] = {xi, hi, xi};
    z[i] = min (z, s[sn].z);
}
```