

DAY 1



ADVANCE PROBLEM 1



ทำง้ให้เขียนโค้ดไ้ขึ้น

ทำง้ให้เขียนโค้ดไวขึ้น



- ใช้ `include<bits/stdc++.h>` ดึง library มาทีเดียว
- ใช้ STL พื้นฐานแบบ `vector`, `map`, `pair`, `stack`, `queue`, `priority queue` ไปเลย
 - ถ้าต้องการ array ที่มี range กว้าง แต่ใช้แค่บางช่อง ให้ใช้ `map`
 - เขียน `sort function` ของ `priority queue` ให้เป็น (ตัวนี้เครื่องหมายกลับ)

```
bool operator <(const data & d2)const{  
    return d2.w > w ;  
}
```

- ถ้า Algo ถูกแต่ไม่ได้เต็ม ให้เช็ค..
 - เปลี่ยน `int` เป็น `long long`
 - โจทย์อาจไม่ให้ตอบค่าที่ได้มาตรงๆ (EX:สถานที่ศักดิ์สิทธิ์ (Sacred Places))
 - ถ้าใช้ `vector` แล้ว memory เกิน ลองใช้ `array` หรือ `map` แทน
 - ถ้าใช้ `map` ลองใช้ `multiple map` เพื่อต้องเก็บหลายตัว
 - อาจต้อง optimize code ให้รันไวขึ้น

เก็บ SUBTASK



ถ้าพบวิธีแก้ปัญหทั้งหมด ตรวจสอบให้แน่ใจก่อนว่าวิธีแก้มันถูก ไม่งั้นอาจจะเสียเวลา 40 นาที++ กับคำตอบที่ไม่ใช่

- โจทย์สามารถ assume ได้ใหม่วิธีของเราจะใช้ได้
 - กราฟมี cycle ได้ไหม / กราฟมี negative edge ไหม
 - Greedy เลือกทางนี้จะดีที่สุดทุกครั้งเลยหรือเปล่า
 - ลองให้ parameter มีค่ามากมาย
 - ลองหา worst case ของโจทย์ว่ามันจะมี input แย่สุดๆได้ขนาดไหน

ถ้ายังคิด ไม่ออก ให้เริ่มจากการเก็บคะแนนบางส่วนไปก่อน

จากการ brute force หรือใช้ algorithm ที่ Big O อาจมากแต่ได้คะแนน



ทำลูกฟื้ต๋อย2 [10 min]

เฉลย ทำลูกพี่ต่อย2



$$\sum_{k=1}^n \binom{n}{k} (Pk)^{Pk} (n-k+1)^{n-k-1} \mod P$$

เดี๋ยว proof อันนี้ออกมาจะได้ 0

ชุดทดสอบย่อย

1. 20% ($1 \leq n \leq 10$)
2. 20% ($1 \leq n \leq 10^3$)
3. 60% ($1 \leq n \leq 10^{18}$)

subtask 1,2 เขียน recursive ตรงๆไปได้
แต่ต้องเขียน function power เอง

subtask 3 ต้องใช้ O แบบ linear
ถามปืบตอบปืบ
แปลว่า input ต้องมี pattern ที่แกะได้

ตัวอย่างชุดทดสอบ

ข้อมูลนำเข้า

1

1 1

ข้อมูลส่งออก

0

บางที !!!

คำตอบก็อยู่ในตัวอย่างที่เขาให้มา
อาจจะเอาคำตอบมาหา pattern

การ observe

แต่เขาก็เสี่ยงจะโดนยิงได้
หากมันดู reverse engineering

เฉลย ทำลูกพี่ต่อຍ2 (40/100)



```
11 f(11 n){
    11 ans=1, N=n;
    while(N>1){
        ans = ans*N;
        N--;
    }
    return ans;
}
```

```
11 power(11 a,11 b){
    11 ans=0;
    while(b--){
        ans = ans*a;
    }
    return ans;
}
```

```
11 sub(11 n,11 p){
    11 ans=0;
    for(11 k=1;k<=n;k++){
        ans += ((11)(f(n)/f(n-k)/f(k))* (11)power(p*k,p*k) * (11)pow(n-k+1,n-k-1))%p;
    }
    return ans;
}
```

```
5
10 20
0
30 56
0
111 11
0
13 13
0
13 29
0
```

แล้วเราเอาโปรแกรมนี้มา observe ต่อได้



$$\sum_{k=1}^n \binom{n}{k} \underline{(Pk)^{Pk}} (n-k+1)^{n-k-1} \mod P$$

จัดรูปได้ $(k \cdot P) \% P = 0$ ทุกครั้ง
ตอบ 0 ทุกรอบ ได้เลย



CC167 M-Prime
[10 min]

เฉลย M-PRIME



จำนวนเฉพาะ คือ จำนวนเต็มบวกที่มีตัวหารที่เป็นบวกอยู่ 2 ตัว คือ 1 กับตัวมันเอง ซึ่งจะมีจำนวนเฉพาะอยู่กลุ่มหนึ่งที่มีคุณสมบัติพิเศษ คือ เป็นจำนวนเฉพาะที่อยู่ในรูป $2^n - 1$ เมื่อ n เป็นจำนวนนับ

บรรทัดแรก รับจำนวนเต็มบวก Q ($1 \leq Q \leq 100,000$) แทนจำนวนคำถาม
 Q บรรทัดถัดมา แต่ละบรรทัดรับจำนวนเต็ม X ($1 \leq X \leq 2^{63} - 1$) สำหรับแต่ละคำถาม

ต้องใช้ O แบบ linear
ถามปุ๊บตอบปั๊บ
จากการที่คำตอบว่า YES มีแค่ 63 แบบ
เราก็ gen มันมาไว้ก่อนได้

อันนี้โค้ดที่เราดูว่ามี $2^{**}n - 1$ ไหนเป็น prime บ้าง
แล้วก็แค่ set map ของค่าเหล่านั้นว่า YES ก็พอ

```
void pre(){
    // X is very big, we can't check by loop
    // obsearb 1 : 2**63 - 1 requair only 64-size array to remember

    unsigned long long temp;
    int n=1;
    while(n<64){
        temp = pow(2,++n) - 1;
        if(!ifPrime(temp)) continue;
        cout << temp << "\n";
    }
}
```



GREEDY PROBLEM

GREEDY PROBLEM



หนึ่งในการเก็บ subtask หรือแก้ปัญหาก็ดี คือการพยายามเลือกไปเส้นทางที่น่าจะดีที่สุด

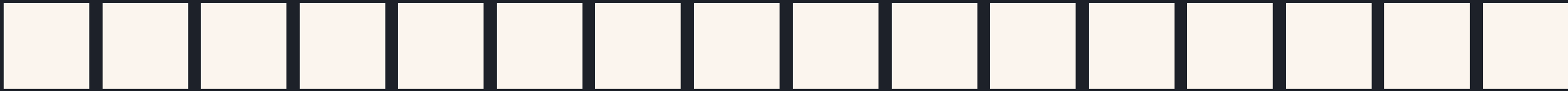
การ greedy ที่พบใน TOI อาจจะได้ไม่ได้โผล่มาเป็นโจทย์ของตัวเองโดยตรง แต่บางครั้งก็ไปเกาะอยู่กับพวก graph หรือ dynamic ก็ได้



CC170 Cupid
[20 min]



" หากคนหนึ่งอยู่ในเมืองที่ i เขาจะมี ความรักได้เฉพาะคนที่อยู่ในเมือง $i - K$ ถึง $i + K$ เท่านั้น "



Best Choice : เลือกคู่จากเมืองที่อยู่ใกล้เมืองของคนนั้นมากที่สุด

Sol : ให้ ช-ญ จับคู่ในเมืองกันเองก่อน ถ้าไม่ครบคู่ก็ให้ไป หาจากเมืองลำดับที่ $i+1$



CC131 Christmas Present

[10 min]

เฉลย CHRISTMAS PRESENT



กรณีที่เป็นไปได้

1. ชื่อจากเมือง A ทั้งสองคู่
2. ชื่อจากเมือง B ทั้งสองคู่
3. ชื่อจากเมือง A และ B อย่างละคู่



ภารกิจ (mission) [20 min]

เจลย ภารกิจ (MISSION)



- สามารถเลือกทำภารกิจไหนก็ได้ แต่ละภารกิจไม่มีบทลงโทษว่าหากไม่ทำจะเสียเท่าไร
 - ควรเลือกภารกิจที่ $B - 2*A$ มาก
- หากทำ i ภารกิจ จะเสียคะแนน $(N-i)**2$
 - แต่อันนี้ assume ไม่ได้ว่าควรเลือกทำ N ภารกิจเสมอไป เพราะ $B-2A$ อาจจะทำให้ค่าติดลบ

Sol : เรียงภารกิจที่ให้คะแนนมากไปน้อย แล้ว loop เลือกทำ 0 ถึง N ภารกิจ เทียบหาจุดที่ได้คะแนนมากที่สุด



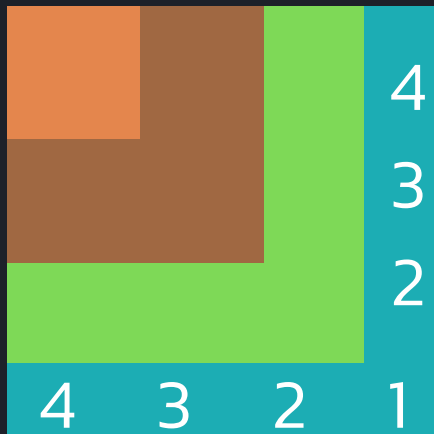
USACO Cow Tipping

[30 min]

เฉลย USACO COW TIPPING



point 1 : ยิง i, j มีมาก(อยู่ใกล้มุมล่างขวา) จัตุรัสทางบนซ้ายที่มาถึงตัวมันและมีผลกระทบต่อมัน จะมีน้อย
ดังนั้น เราควรพลิกวัวจาก $i=N-1, j=M-1$ ก่อน แล้วค่อยๆไปทางซ้าย เพื่อที่จะให้เกิดการพลิกซ้ำซ้อนน้อยสุด





SWEEP LINE

SWEEP LINE



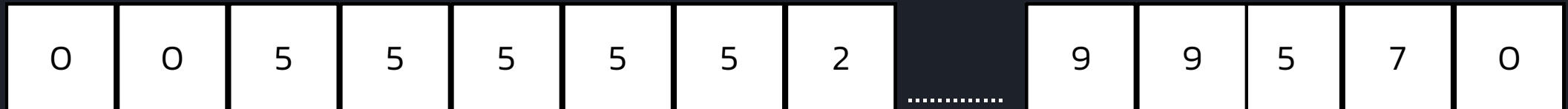
SWEEP LINE -> เส้นกวาด -> การคำนวณแบบกวาด โดยที่จะเปลี่ยนค่าแค่จุดสำคัญ



step 1 : เก็บเฉพาะจุดที่ค่ามีการเปลี่ยนแปลง (มักเก็บใน map)



step 2 : ใช้ iterator ของ map เพื่อหาค่าของแต่ละช่อง หรือหาค่า min/max ของทั้งหมด



SWEEP LINE



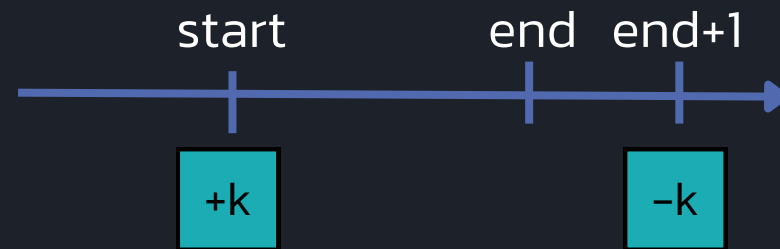
```
int main(){
    ios_base::sync_with_stdio(0), cin.tie(0);

    cin >> N;
    for( int i=0 ; i<N ; i++){
        cin >> s >> f;
        m[s]+=1, m[f+1]-=1;
        NX=max(NX,f);
    }

    int data[N], temp=0;
    for(int i=1;i<NX;i++){
        temp+=m[i];
        data[i]=temp;
        cout << data[i] << " ";
    }

    return 0;
}
```

step 1 : เก็บเฉพาะจุดที่ค่ามีการเปลี่ยนแปลง



step 2 : หาค่าของแต่ละช่อง โดยเก็บค่าที่กำลังเปลี่ยนแปลงไปเรื่อยๆด้วยตัวแปร temp



CSES Restaurant Customers

[10 min]

เจलय CSES RESTAURANT CUSTOMERS



Sweep Line ตรงๆเลย

ใช้ map มา mark เวลาเข้า-ออก แล้ว for auto หาจุดที่คนเยอะที่สุด



กิ้งก่า (iguana) [20 min]

เจลาย กิ้งก่า (IGUANA)



ใช้ sweep line มา track ก่อนว่าเด็กมือบอนเริ่มจัมกิ้งก่า-หยุดจัมตรงช่วงไหนบ้าง
แล้ว for loop เพื่อ

- เมื่อมีการจัม ใช้ตัวแปร sd เช็คว่ากิ้งก่าถูกจัมไปที่ครั้ง
- $sd \% 3$ จะได้สีของกิ้งก่า เป็น แดง-เขียว-น้ำเงิน ตามลำดับ
- ใช้ตัวแปร ct[] เก็บว่ากิ้งก่าสีไหนมีกี่ตัว

แล้วค่อยมาดูว่าสีอาหารแต่ละสีต้องไปจัมกิ้งก่าที่ครั้ง

HOMework



- tumso18 mathmath
- 1090 หางไก่ (cocktail)
- CSES Ferris Wheel

จุดผิดของข้อ math math นี้คือเลขมันใหญ่มากๆ เขียนโปรแกรม observe จะได้ค่าไม่ตรง ให้ใช้เว็บนี้แทน
<https://defuse.ca/big-number-calculator.htm>