

강의 개관

알고리즘을 공부하는 이유

Dosawas 2024-02-13

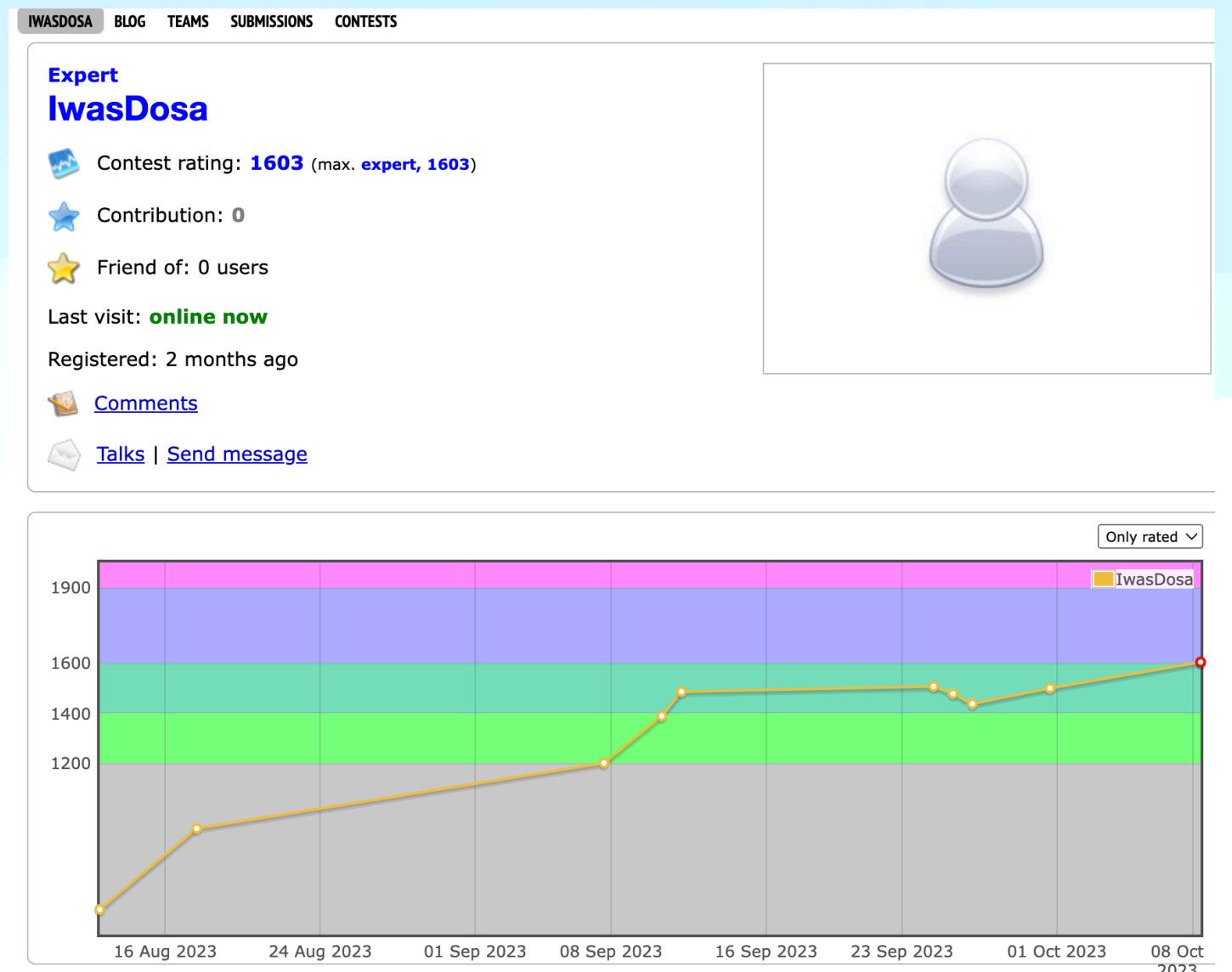
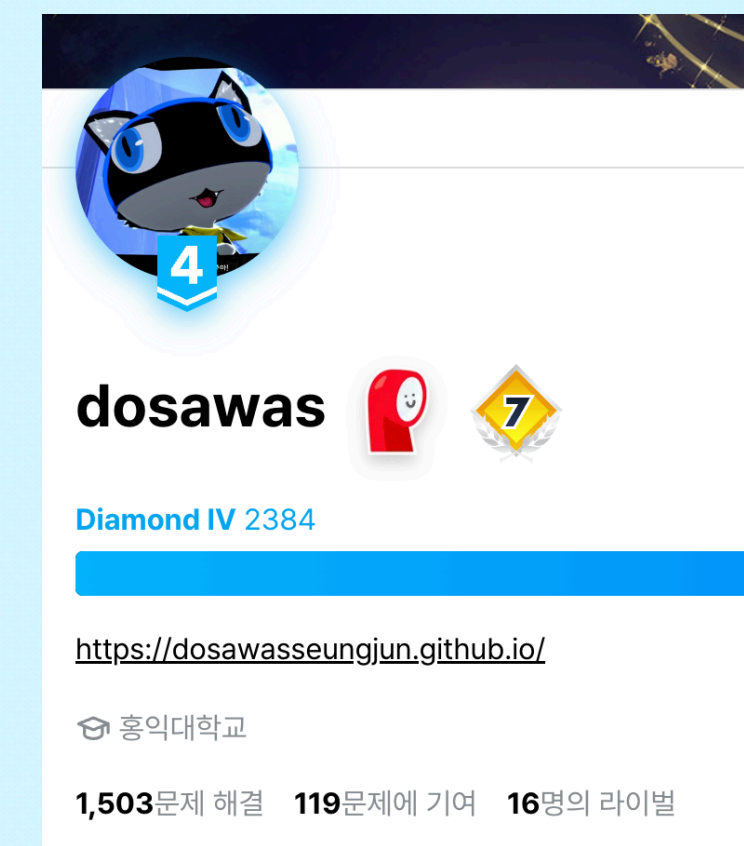
인사 및 차례

차례

- 1. 강사소개
- 2. 강의 방식
- 3. 알고리즘을 공부하는 이유
- 4. 라이브러리 세팅
- 5. 간단한 문제 풀이 + 정렬, 그리디

강사 소개

- 이름, 학번, 학과 : 한승준, 19학번, 자율전공(컴퓨터공학 진입)
- 백준 handle : dosawas
- 경력
 - 백준 1502문제 해결, 다이아 4 (solved.ac 기준)
 - 코드포스 Blue
 - 각종 대회
 - IcpC-sinchon algorithm camp - 7등(22summer), 3등(23winter), 2등(23summer)
 - Suapc 2023 summer, 2024 winter 10등 (dosawas, smjun, mastershim)
 - 2022 홍익대학교 컴퓨터공학과 프로그래밍 경진대회 은상 (3위)



강의 방식

문제를 푸는 방법

- 1. 각 알고리즘이 언제 사용되는지
 - Ex) 2의 거듭제곱
- 2. 그 알고리즘의 원리
 - Ex) 2진수꼴의 원리
- 3. 문제 풀이
 - 예외가 있을까?
 - 시간복잡도는 얼마나 될까?
- 질문 : 모르겠으면 물어보는 것이 도움이 많이 된다.
 - (저는 c++, python 둘다 합니다. java는 좀 못함...)

저는 알고리즘을 자세히 가르치지 않고
문제풀이(PS)와 대회(CP)
를 잘하는 법을 가르칩니다.

```
x){  
0;  
x - 1)) == 0);  
  
tt;  
  
>> x;  
er(x) cout << "YES\n";  
else cout << "NO\n";  
}  
}
```


알고리즘을 공부하는 이유

1. 효율

- 예를 들어, 교수님이 학부연구생 A와 B에게 다음과 같은 일을 시킨다고 해보자.
- 교수 : “학생들 과제를 다 출력해서 학번 순으로 정렬해서 가져다 주게”
- 이 때, 과제들의 파일명이 제멋대로지만 학번은 모두 기입되어 있는 상황입니다.



학생 A :
헉 빨리 일 해야지...!!
일단 출력해서 정렬 작업해야겠다!
학생이 거의 200명이니까... 3시간은
잡아야겠는걸

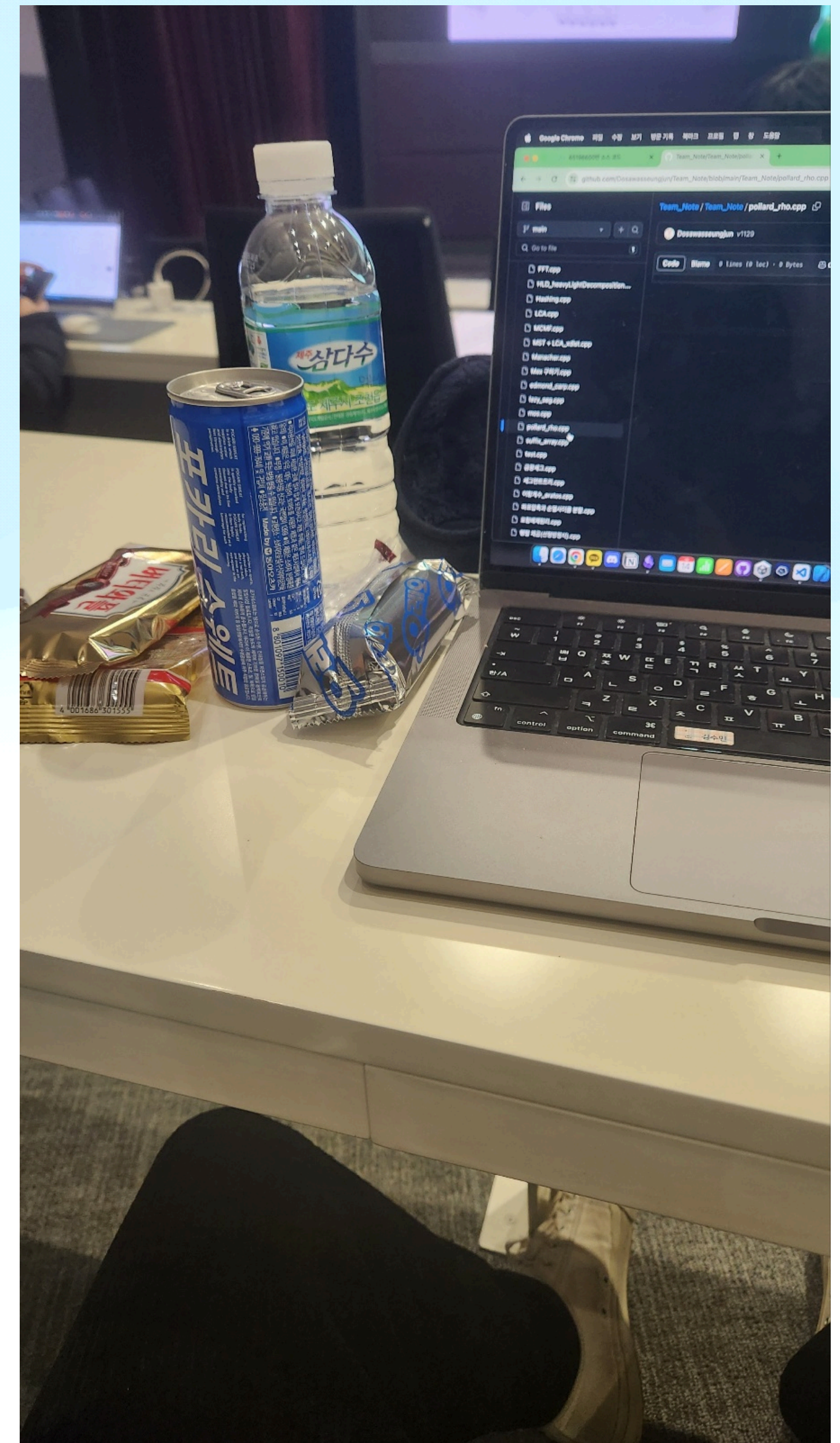


학생 B :
학번순으로 정렬을 컴퓨터로 한
다음에 순서대로 출력하면 끝나겠군~
30분이면 끝나겠어 ㅎㅎ

알고리즘을 공부하는 이유

2. 재미

- 각종 대회를 참가하는 일은 정말 즐겁습니다.
- 맛있는 간식도 줍니다 (사진 자료)
- 어려운 문제를 토론하는 것은 생산적이고 재밌습니다.
- 똑똑한 친구들을 사귄 수 있습니다.



알고리즘을 공부하는 이유

3. 회사가 좋아함

- 후원사

- 현대모비스, 현대 오토에버

- 넥슨

- 삼성 전자, LG 전자

- 코딩 테스트를 보는 각종 IT 회사들

- 대회를 직접 주최하는 회사도 많음(구글, 페이스북, 삼성, 현대)



ucpc를 후원하는 후원사 목록(<https://2023.ucpc.me/sponsor/>)

라이브러리 세팅

내가 쓰는 라이브러리와 템플릿

2024-HiarcStudy > 1_ > C++ Mytemplate.cpp > ...

```
1  #include <bits/stdc++.h>
2  #define fast_io cin.tie(NULL); ios_base::sync_with_stdio(false);
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef pair<ll, ll> pll;
7  typedef tuple<int, int, int> tiii;
8  typedef tuple<ll, ll, ll> tlll;
9  #define xx first
10 #define yy second
11
12 int main(){
13     fast_io
14 }
```

← 라이브러리

← 빠른 입출력

← typedef

← define

문제 풀이

어떻게 문제에 접근해야할까?

- 제가 생각한 가장 중요한 요소
 - 문제 이해
 - 문제 풀이법 생각 (test data 따라가보기, 수식으로 써보기 등)
 - 반례, 예외 찾기
- 코드 짜면서 생각하기 vs. 생각하고 코드짜기

간단한 문제 풀이

1. 관찰, 수학

- 문제를 풀 때는 관찰하는 것이 가장 중요합니다.
- [30021 순열 선물하기](<https://www.acmicpc.net/problem/30021>)

문제

준원이는 정수 $1, 2, \dots, N$ 을 가지고 있다. 준원이는 자신이 가지고 있는 정수를 하루에 하나씩 N 일에 걸쳐서 현석이에게 선물해 주려고 한다. 그런데, 현석이는 소수를 싫어하기 때문에 현석이가 지금까지 선물 받은 수의 합이 소수가 되는 날이 없도록 해야 한다.

준원이가 현석이를 만족시키면서 정수 $1, 2, \dots, N$ 을 선물하려면 어떤 순서로 선물해야 하는지 구해보자.

입력

첫째 줄에 정수 N ($1 \leq N \leq 5\,000$)이 주어진다.

간단한 문제 풀이

2. 정렬

- 여기서는 정렬 알고리즘이 무엇이 있는지는 배우지 않습니다. (Ex) 버블, 삽입, merge
- `sort(arr.begin(), arr.end())` 를 사용하는 경우만 익힙시다.
 - c++의 `vector` 컨테이너는 자주 사용해서 익히는게 좋습니다.
- 정렬 : 자료에 순서라는 정보를 추가 -> 쿼리에 유용 (Kth 수 찾기)
- 순서가 필요할 때 - [K번째 수](<https://www.acmicpc.net/problem/11004>)
- 단조로운(monotone) 정보가 필요할 때

간단한 문제 풀이

2. 정

역원소

5 실버

시간 제한

1 초

문제

모든 원소

단, 원소를

입력

첫 번째로

```
13 int main(){
14     fast_io
15     int n; cin >> n;
16     vector<ll> num(n);
17     for(int i=0;i<n;i++){
18         string s; cin >> s;
19         reverse(s.begin(), s.end());
20         num[i] = stoll(s);
21     }
22     sort(num.begin(), num.end());
23     for(ll x : num) cout << x << '\n';
24 }
```



한국어 ▼

정답 비율

43.091%

단, 입력하는 정수는 10^{14} 을 넘어선 안 됩니다.

도 들어갈 수 있습니다.

간단한 문제 풀이

3. 그리디

선 긋기 성공



5 골드 V

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	192 MB	22995	8323	6042	35.280%

문제

매우 큰 도화지에 자를 대고 선을 그으려고 한다. 선을 그을 때에는 자의 한 점에서 다른 한 점까지 긋게 된다. 선을 그을 때에는 이미 선이 있는 위치에 겹쳐서 그릴 수도 있는데, 여러 번 그은 곳과 한 번 그은 곳의 차이를 구별할 수 없다고 하자.

이와 같은 식으로 선을 그었을 때, 그려진 선(들)의 총 길이를 구하는 프로그램을 작성하시오. 선이 여러 번 그려진 곳은 한 번씩만 계산한다.

입력

첫째 줄에 선을 그은 횟수 N ($1 \leq N \leq 1,000,000$)이 주어진다. 다음 N 개의 줄에는 선을 그을 때 선택한 두 점의 위치 x, y ($-1,000,000,000 \leq x < y \leq 1,000,000,000$)가 주어진다.

간단한 문제 풀이

4. 실수 하지말자

- Integer Overflow를 조심하자!! Ex) $1 \ll 60$, shift 연산의 함정
- [26008 해시해킹](<https://www.acmicpc.net/problem/26008>)
 - 문제 이해는 꽤 까다롭지만 문제를 잘 이해한다면 M^{N-1} 을 $1e9+7$ 로 나눈 나머지를 구하면 됩니다.
 - 이 문제는 홍익대 프로그래밍 대회 문제였음.
 - 제가 옛날에 했던 실수 : $N - 1$ 이 0인 경우를 처리 안함....

마무리

재미 붙이기와 꾸준함

- 알고리즘을 잘하는 방법은 꾸준히 **재미있게** 하는 것이라고 생각합니다.
- 여러분들도 재미를 붙이신다면 적어도 코딩테스트는 걱정이 없어집니다.

