

제 1회 전국 대학생 프로그래밍 대회 연합 동아리 대회 풀이

Algospot.com 출제진 일동

2011년 8월 13일

감사합니다 > . <



A - 사각형 그리기

- Submissions : 25
- Accepted : 21 (84%)
- First Team/Time : Extra-1 in 2 mins
- 출제 : LIBe (정현환)
- 검수 : xhae (류현종)
- Category : 구현

A - 사각형 그리기

더이상의 자세한 설명은 생략합니다.

A - 사각형 그리기

더이상의 자세한 설명은 생략....이 아니고

- 적절한 if문으로 짜세요..

A - 사각형 그리기

더이상의 자세한 설명은 생략....이 아니고

- 적절한 if문으로 짜세요..

- $x_4 = x_1 \oplus x_2 \oplus x_3$

$$y_4 = y_1 \oplus y_2 \oplus y_3$$

B - Wooksinness of a Graph

- Submissions : 55
- Accepted : 12 (22%)
- First Team/Time : SNU-1 (12 mins)
- 출제 : astein (김진호)
- 검수 : LIBe (정현환)
- Category : 그래프

B - Wooksinness of a Graph

- $|V| \leq 20$ 이면 impossible

B - Wooksinness of a Graph

- $|V| \leq 20$ 이면 impossible
- $|E| = 00$ 이면 : 3

B - Wooksinness of a Graph

- $|V| \leq 20$ 이면 impossible
- $|E| = 0$ 이면 : 3
- 이미 싸이클이 있으면 : 0
 - 당연한 거지만 이 cycle의 길이는 3보다 훨씬 클 수도 있어용

B - Wooksinness of a Graph

- $|V| \leq 20$ 이면 impossible
- $|E| = 0$ 이면 : 3
- 이미 싸이클이 있으면 : 0
 - 당연한 거지만 이 cycle의 길이는 3보다 훨씬 클 수도 있어용
- (싸이클이 없고) 연결된 3개의 정점이 있으면 : 1

B - Wooksinness of a Graph

- $|V| \leq 20$ 이면 impossible
- $|E| = 0$ 이면 : 3
- 이미 싸이클이 있으면 : 0
 - 당연한 거지만 이 cycle의 길이는 3보다 훨씬 클 수도 있어용
- (싸이클이 없고) 연결된 3개의 정점이 있으면 : 1
- 나머지 : 2

E - 화이트 칼라

- Submissions : 77
- Accepted : 10 (13%)
- First Team/Time : Sogang-1 (24 mins)
- 출제 : albertain (이태윤)
- 검수 : LIBe (정현환)
- Category : 최단 경로

E - 화이트 칼라

- 0번에서 출발하는 최단경로, $n - 1$ 번에서 출발하는 최단경로를 구한다
 - Dijkstra's Algorithm
 - BFS..?

E - 화이트 칼라

- 0번에서 출발하는 최단경로, $n - 1$ 번에서 출발하는 최단경로를 구한다
 - Dijkstra's Algorithm
 - BFS..?
- find v such that $d(0, v) + d(v, n - 1) = d(0, n - 1)$

C - Run Run Run

- Submissions : 34
- Accepted : 4 (12%)
- First Team/Time : SKKU-1 (165 mins)
- 출제 : albertain (이태윤)
- 검수 : LIBe (정현환)
- Category : 상태공간 탐색 (최단경로)

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데
- 당근을 먹으면 하루에 몇 칸 더 움직일 수 있다
- 며칠 걸리나?

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데
- 당근을 먹으면 하루에 몇 칸 더 움직일 수 있다
- 며칠 걸리나?

상태공간 탐색

- state : (r, c, v) 약 $1000 \times 1000 \times 5$

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데
- 당근을 먹으면 하루에 몇 칸 더 움직일 수 있다
- 며칠 걸리나?

상태공간 탐색

- state : (r, c, v) 약 $1000 \times 1000 \times 5$
- 하루걸리는 평범한 이동 (cost : 1)

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데
- 당근을 먹으면 하루에 몇 칸 더 움직일 수 있다
- 며칠 걸리나?

상태공간 탐색

- state : (r, c, v) 약 $1000 \times 1000 \times 5$
- 하루걸리는 평범한 이동 (cost : 1)
- 당근 먹기 ($v = 0$ 일때) (cost : 0)

문제

- knight 처럼 움직일 수 있는 말이 어디서부터 어디까지 이동하는데
- 당근을 먹으면 하루에 몇 칸 더 움직일 수 있다
- 며칠 걸리나?

상태공간 탐색

- state : (r, c, v) 약 $1000 \times 1000 \times 5$
- 하루걸리는 평범한 이동 (cost : 1)
- 당근 먹기 ($v = 0$ 일때) (cost : 0)
- 당근 소모하며 움직이기 (cost : 0)

C - Run Run Run

BFS !!!!!!!!!!

BFS ...를 하면 망하고...

BFS ...를 하면 망하고...

- priority queue를 써서 탐색하거나 (i.e. dijkstra)

BFS ...를 하면 망하고...

- priority queue를 써서 탐색하거나 (i.e. dijkstra)
- deque를 써서 cost가 0일 때는 앞에 넣고, cost가 1일 때는 뒤에 넣는 식으로 탐색하면 Accepted!

BFS ...를 하면 망하고...

- priority queue를 써서 탐색하거나 (i.e. dijkstra)
- deque를 써서 cost가 0일 때는 앞에 넣고, cost가 1일 때는 뒤에 넣는 식으로 탐색하면 **Accepted!**

Tricky Points

- 이동 시 8방향 말고 가만히 있는것도 처리해줘야...

H - 맑은 하늘 프로젝트

- Submissions : 25
- Accepted : 6 (24%)
- First Team/Time : Ajou-1 (47 mins)
- 출제 : domeng (이도경)
- 검수 : JongMan (구종만)
- Category : DP

H - 맑은 하늘 프로젝트

- 모든 레이저포는 왼쪽에서부터 쏘는 것이 이득이다.

H - 맑은 하늘 프로젝트

- 모든 레이저포는 왼쪽에서부터 쏘는 것이 이득이다.
- 레이저포를 쏘는 좌표는 항상 어떤 한 구름의 왼쪽 끝 좌표만으로도 충분하다.

H - 맑은 하늘 프로젝트

- 모든 레이저포는 왼쪽에서부터 쏘는 것이 이득이다.
- 레이저포를 쏘는 좌표는 항상 어떤 한 구름의 왼쪽 끝 좌표만으로도 충분하다.
- Dynamic Programming!

H - 맑은 하늘 프로젝트

- $D(n, k) := 1 \cdots n$ 번째 구름까지 k 번 발사해서 전부 제거했을 때 필요한 최소비용

H - 맑은 하늘 프로젝트

- $D(n, k) := 1 \cdots n$ 번째 구름까지 k 번 발사해서 전부 제거했을 때 필요한 최소비용
- $D(n, k)$ 를 계산하기 위해 $D(m, k - 1)$ (단, $m < n$)을 찾는다.

H - 맑은 하늘 프로젝트

- $D(n, k) := 1 \cdots n$ 번째 구름까지 k 번 발사해서 전부 제거했을 때 필요한 최소비용
- $D(n, k)$ 를 계산하기 위해 $D(m, k - 1)$ (단, $m < n$)을 찾는다.
- $O(n^2k)$

H - 맑은 하늘 프로젝트

- $D(n, k) := 1 \cdots n$ 번째 구름까지 k 번 발사해서 전부 제거했을 때 필요한 최소비용
- $D(n, k)$ 를 계산하기 위해 $D(m, k - 1)$ (단, $m < n$)을 찾는다.
- $O(n^2k)$
- <출제자> $O(nk)$ 로 줄일수 있을것도 같지만 출제자도 생각하기 귀찮아서 안했다.

I - Turn Off The Lights

- Submissions : 23
- Accepted : 1 (4%)
- First Team/Time : SNU-1 (88 mins)
- 출제 : wook (최종욱)
- 검수 : LIBe (정현환)
- Category : DP (2^n state)

I - Turn Off The Lights

$f(i, cmask)$: i 번째 행까지를 모두 끄기 위해 필요한 최소 연산의 수. 단 $cmask$ 는 각 column이 눌리고 있는지 아닌지를 나타냄. ($0 \leq i < n, cmask \in \{0, 1\}^m$)

- 이 줄을 뒤집는 데 필요한 group의 수를 세고(행연산), 열 연산을 $O(m)$ 으로 센다.
- $O(n2^m m)$

I - Turn Off The Lights

왼쪽 위에서부터 차례대로 결정해 나갑니다.

I - Turn Off The Lights

왼쪽 위에서부터 차례대로 결정해 나갑니다.

$f(i, j, cmask, r)$: (i, j) 와 같은행 오른쪽에 있거나 (i, j) 보다 아래에 있는 칸을 다 끄기 위한 동작의 최소횟수
단, $cmask$ 는 각 column이 눌리고 있는지 아닌지를,
 r 은 i 번 행이 눌리고 있는지 아닌지를 나타냄.
 $(0 \leq i < n, 0 \leq j < m, cmask \in \{0, 1\}^m, r \in \{0, 1\})$

I - Turn Off The Lights

왼쪽 위에서부터 차례대로 결정해 나갑니다.

$f(i, j, cmask, r)$: (i, j) 와 같은행 오른쪽에 있거나 (i, j) 보다 아래에 있는 칸을 다 끄기 위한 동작의 최소횟수
단, $cmask$ 는 각 column이 눌리고 있는지 아닌지를,
 r 은 i 번 행이 눌리고 있는지 아닌지를 나타냄.
 $(0 \leq i < n, 0 \leq j < m, cmask \in \{0, 1\}^m, r \in \{0, 1\})$

- 최종 (i, j) 칸의 상태 : $cmask_j \oplus r \oplus a_{ij}$

I - Turn Off The Lights

왼쪽 위에서부터 차례대로 결정해 나갑니다.

$f(i, j, cmask, r)$: (i, j) 와 같은행 오른쪽에 있거나 (i, j) 보다 아래에 있는 칸을 다 끄기 위한 동작의 최소횟수
단, $cmask$ 는 각 column이 눌리고 있는지 아닌지를,
 r 은 i 번 행이 눌리고 있는지 아닌지를 나타냄.
 $(0 \leq i < n, 0 \leq j < m, cmask \in \{0, 1\}^m, r \in \{0, 1\})$

- 최종 (i, j) 칸의 상태 : $cmask_j \oplus r \oplus a_{ij}$ 이 값을 0으로 만들어야 함
- 각 경우마다 적절히 $cmask_j$ 또는 r 를 바꿔주고 $(i, j + 1)$ 로 이동.
- $O(nm2^m)$

J - Lottery Games

- Submissions : 20
- Accepted : 2 (10%)
- First Team/Time : Extra-1 (262 min)

- 출제 : ldtl (이후연)
- 검수 : LIBe (정현환)
- Category : 수학(확률)

J - Lottery Games

각 티켓의 winning probability를 구하면 된다.

각 티켓의 winning probability를 구하면 된다.

- $$p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$$

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA
 - Fraction (분수 구현) :

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA
 - Fraction (분수 구현) : WA

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA
 - Fraction (분수 구현) : WA
 - Fraction with BigInteger : AC

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA
 - Fraction (분수 구현) : WA
 - Fraction with BigInteger : AC
 - 64bit integer용 곱셈, 128bit integer용 덧셈만 구현해도 AC

각 티켓의 winning probability를 구하면 된다.

- $p(n, m, k) = \frac{\sum_{i=k}^m \binom{m}{i} \times \binom{n-m}{m-i}}{\binom{n}{m}}$
- 그런데 Precision Error!
 - double : WA
 - Fraction (분수 구현) : WA
 - Fraction with BigInteger : AC
 - 64bit integer용 곱셈, 128bit integer용 덧셈만 구현해도 AC
- $\binom{50}{25} = 126,410,606,437,752$

- Submissions : 13
- Accepted : 0 (0%)
- First Team/Time : N/A
- 출제 : ryuwonha (류원하)
- 검수 : JongMan (구종만), Astein (김진호)
- Category : 그리디

검수위원의 말! 말! 말!

- <JM> 이게 350^1 이라고 한번더 주장해보시지
- <LiBe> 새벽에 20대 후반의 한 남자를 울며 잠들게 한 문제
- <JM> 나 다시는 빙꺼 검수안해
- <astein> 저 G 한번에 맞았음 > . <
- <astein> 짜증나 죽는 줄 알았음..

¹(출제자가 주장했던) Topcoder SRM Div1 기준 난이도

방법 1은 쉬우니까 떠올리고나면 문제에서 헤어나올 수 없..
기를 바랬어요

방법 1은 쉬우니까 떠올리고나면 문제에서 헤어나올 수 없..
기를 바랬어요

- 자명하게 일단 금메달을 다 먹고 시작하고,
- 남은 $2K$ 개의 메달을 한 팀에 최대 K 개씩 분배해야 하는데,

방법 1은 쉬우니까 떠올리고나면 문제에서 헤어나올 수 없..
기를 바랬어요

- 자명하게 일단 금메달을 다 먹고 시작하고,
- 남은 $2K$ 개의 메달을 한 팀에 최대 K 개씩 분배해야 하는데,
- 나보다 이미 잘한 사람들은 메달을 아무리 줘도 상관없으니까 나눠주고
- 남는 것들을 나머지 팀에 나눠줄 수 있으면 끝내고..

방법 1은 쉬우니까 떠올리고나면 문제에서 헤어나올 수 없..
기를 바랬어요

- 자명하게 일단 금메달을 다 먹고 시작하고,
- 남은 $2K$ 개의 메달을 한 팀에 최대 K 개씩 분배해야 하는데,
- 나보다 이미 잘한 사람들은 메달을 아무리 줘도 상관없으니까 나눠주고
- 남는 것들을 나머지 팀에 나눠줄 수 있으면 끝내고..
- 그렇지 않으면 나머지 팀들 중에서 제일 잘 한 애한테 K 개를 몰아주면 됩니다.
- 다시 남는 것들을 나머지 팀에 나눠줄 수 있으면 끝내고..
- 그렇지 않으면 두 팀한테 질 수밖에 없겠죠.

방법 2의 경우 자명하게 금메달을 다 먹은 다음에,

- 남은 팀 중 나보다 절대 앞설 수 없거나 이미 앞선 애들을 제거하고,
- 그런 애들이 2팀 이상이면 은메달과 동메달을 K 개씩 줘도 순위에 변동이 없으니 주고 끝내고,

방법 2의 경우 자명하게 금메달을 다 먹은 다음에,

- 남은 팀 중 나보다 절대 앞설 수 없거나 이미 앞선 애들을 제거하고,
- 그런 애들이 2팀 이상이면 은메달과 동메달을 K 개씩 줘도 순위에 변동이 없으니 주고 끝내고,
- 그런 애들이 1팀이면, 은메달을 몰아준다음에

방법 2의 경우 자명하게 금메달을 다 먹은 다음에,

- 남은 팀 중 나보다 절대 앞설 수 없거나 이미 앞선 애들을 제거하고,
- 그런 애들이 2팀 이상이면 은메달과 동메달을 K 개씩 줘도 순위에 변동이 없으니 주고 끝내고,
- 그런 애들이 1팀이면, 은메달을 몰아준다음에
 - 남은 팀들에게 동메달을 나눠주되,
 - 나보다 은메달의 수가 적은 애가 존재하면 절대 앞설 수 없으니 주고 끝내고,
 - 그런 애들이 없으면 동메달을 K 개 골고루 나눠줄 수 있는지 확인하고,
 - 그게 불가능하면 한 팀을 내 앞으로 보내는 수밖에 없죠.

- 그런 애들이 없으면,
 - 은메달을 분배할 수 있는지 보고,
 - 은메달을 넉넉히 분배할 수 있으면 여유있는 팀이 있다는 뜻이니까 거기에 동메달을 몰아줄 수 있고,

- 그런 애들이 없으면,
 - 은메달을 분배할 수 있는지 보고,
 - 은메달을 넉넉히 분배할 수 있으면 여유있는 팀이 있다는 뜻이니까 거기에 동메달을 몰아줄 수 있고,
- 딱 떨어지면
 - 지금까지 나눠준 은메달에 덧붙여 추가로 줄 수 있는 동메달의 수가 넉넉한지 확인하고,
 - 넉넉하면 나눠주고 끝나고
 - 아니면 한 팀한테 질 수밖에 없고

- 그런 애들이 없으면,
 - 은메달을 분배할 수 있는지 보고,
 - 은메달을 넉넉히 분배할 수 있으면 여유있는 팀이 있다는 뜻이니까 거기에 동메달을 몰아줄 수 있고,
- 딱 떨어지면
 - 지금까지 나눠준 은메달에 덧붙여 추가로 줄 수 있는 동메달의 수가 넉넉한지 확인하고,
 - 넉넉하면 나눠주고 끝나고
 - 아니면 한 팀한테 질 수밖에 없고
- 모자라면
 - 한 팀을 집어서 은메달을 몰빵한 다음에
 -

- 그런 애들이 없으면,
 - 은메달을 분배할 수 있는지 보고,
 - 은메달을 넉넉히 분배할 수 있으면 여유있는 팀이 있다는 뜻이니까 거기에 동메달을 몰아줄 수 있고,
 - 딱 떨어지면
 - 지금까지 나눠준 은메달에 덧붙여 추가로 줄 수 있는 동메달의 수가 넉넉한지 확인하고,
 - 넉넉하면 나눠주고 끝나고
 - 아니면 한 팀한테 질 수밖에 없고
 - 모자라면
 - 한 팀을 집어서 은메달을 몰빵한 다음에
 -

알고스팟은 여러분을 사랑합니다!

- Submissions : 8
- Accepted : 0 (0%)
- First Team/Time : N/A
- 출제 : ainu7 (유원석)
- 검수 : astein (김진호), rrx (김재홍)
- Category : 수학

문제를 읽었더니 n 이 10억 !!!

문제를 읽었더니 n 이 10억 !!!

- **간략화 1.** $n \geq 90$ 만인 경우에는 각 노트의 점수가 1점 이하이므로, 모든 점수를 만들 수 있습니다. (비둘기 집의 원리!)

문제를 읽었더니 n 이 10억 !!!

- **간략화 1.** $n \geq 90$ 만인 경우에는 각 노트의 점수가 1점 이하이므로, 모든 점수를 만들 수 있습니다. (비둘기 집의 원리!)
- **간략화 2.** $n \geq 27$ 만인 경우에는 30% 점수가 1점이므로, 27만점 이하의 모든 점수를 만들 수 있습니다. 그 이후의 점수는 70%, 100%의 점수들을 조합해서 90만까지 모두 만드는 것이 가능합니다.

문제를 읽었더니 n 이 10억 !!!

- **간략화 1.** $n \geq 90$ 만인 경우에는 각 노트의 점수가 1점 이하이므로, 모든 점수를 만들 수 있습니다. (비둘기 집의 원리!)
- **간략화 2.** $n \geq 27$ 만인 경우에는 30% 점수가 1점이므로, 27만점 이하의 모든 점수를 만들 수 있습니다. 그 이후의 점수는 70%, 100%의 점수들을 조합해서 90만까지 모두 만드는 것이 가능합니다.

그래서 $n \leq 27$ 만 이라고 가정할 수 있습니다.

기본 아이디어 : 총 노트가 n 개일 때, x 점을 만들자!

기본 아이디어 : 총 노트가 n 개일 때, x 점을 만들자!

- 몇 개의 노트를 쳐야 x 점을 받을 수 있을지를 계산 (단, 0.1 단위까지 체크)

기본 아이디어 : 총 노트가 n 개일 때, x 점을 만들자!

- 몇 개의 노트를 쳐야 x 점을 받을 수 있을지를 계산 (단, 0.1 단위까지 체크)
- 총 270만 개의 경우가 있으므로 적절한 DP로 해결합니다.

기본 아이디어 : 총 노트가 n 개일 때, x 점을 만들자!

- 몇 개의 노트를 쳐야 x 점을 받을 수 있을지를 계산 (단, 0.1 단위까지 체크)
- 총 270만 개의 경우가 있으므로 적절한 DP로 해결합니다.
- DP로 하기 싫으면 경우를 “아주 잘” 나눠서 하면 풀 수 있어요.

D - 재하의 장난감

- Submissions : 14
- Accepted : 1 (7%)
- First Team/Time : SNU-1 (249 min)
- 출제 : ainu7 (유원석)
- 검수 : wook (최종욱), RRx (김재홍)
- Category : 기하 :D

아..... 레알 깊은 백침이
느껴지는 기하문제다.....

D - 재하의 장난감

- 평면 그래프 상에서 '면'의 개수를 세는 문제.

D - 재하의 장난감

- 평면 그래프 상에서 '면'의 개수를 세는 문제.
- Euler's Characteristic Formula : $v - e + f = 2$

D - 재하의 장난감

- 평면 그래프 상에서 '면'의 개수를 세는 문제.
- Euler's Characteristic Formula : $v - e + f = 2$
- n 개의 선분을 pairwise 교점을 구해 점의 수 v 를 센다.

D - 재하의 장난감

- 평면 그래프 상에서 '면'의 개수를 세는 문제.
- Euler's Characteristic Formula : $v - e + f = 2$
- n 개의 선분을 pairwise 교점을 구해 점의 수 v 를 센다.
- 각 점마다 인접한 선분의 수를 세서 degree의 합으로부터 e 를 구하거나

D - 재하의 장난감

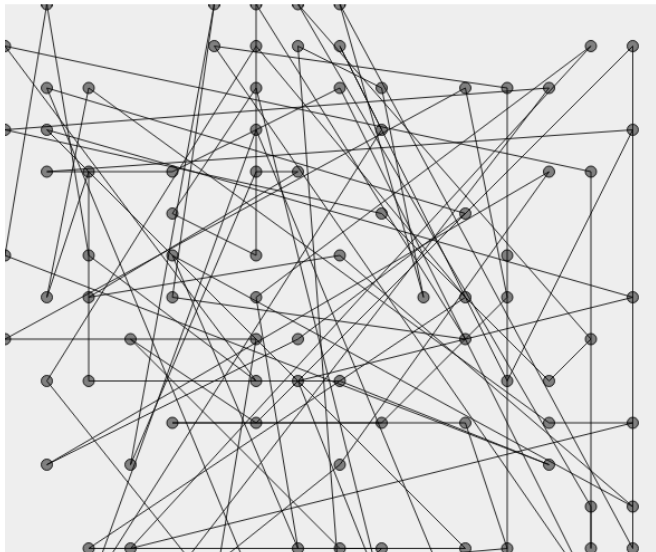
- 평면 그래프 상에서 '면'의 개수를 세는 문제.
- Euler's Characteristic Formula : $v - e + f = 2$
- n 개의 선분을 pairwise 교점을 구해 점의 수 v 를 센다.
- 각 점마다 인접한 선분의 수를 세서 degree의 합으로부터 e 를 구하거나
- 각 직선마다 위에 놓인 점들을 정렬한 뒤 minimal edge를 세주면 e 를 구하면 끝

D - 재하의 장난감

- 평면 그래프 상에서 '면'의 개수를 세는 문제.
- Euler's Characteristic Formula : $v - e + f = 2$
- n 개의 선분을 pairwise 교점을 구해 점의 수 v 를 센다.
- 각 점마다 인접한 선분의 수를 세서 degree의 합으로부터 e 를 구하거나
- 각 직선마다 위에 놓인 점들을 정렬한 뒤 minimal edge를 세주면 e 를 구하면 끝
- $f = 1 - v + e$!!!!!

D - 재하의 장난감

기하의 묘미



수고하셨습니다

여러분 화이팅 > . <