shake!

2019 경인지역 6개대학
연합 프로그래밍 경시대회 – **본선 풀이**

A. 패턴

정답자: ac: 58 / submission: 187

출제자: 홍준표 (아주대학교)

A. 패턴

출제자: 홍준표

- 가장 많은 질문이 있었습니다..

A	{1,6,7}은 가능한 패턴인가요?
↪	문제를 자세히 읽어주세요.
A	1, 8, 3 같은 경우는 유효한 패턴으로 인정되는 건가요? 인접한 패턴이라는게 어떤건지 궁금합니다.
↪	문제를 자세히 읽어주세요.
A	다음패턴으로 갈때 최단경로로만 갈수 있는 건가요?
↪	문제를 자세히 읽어주세요.

- 자세히 읽어달라는 말 외에는 해드릴 수 없어 죄송합니다

shake!

A. 패턴

출제자: 홍준표

- 패턴의 길이는 3 이상이다.
- 패턴을 나타내는 수열에는 같은 점이 두번 이상 등장하지 않는다.
- **수열의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.**

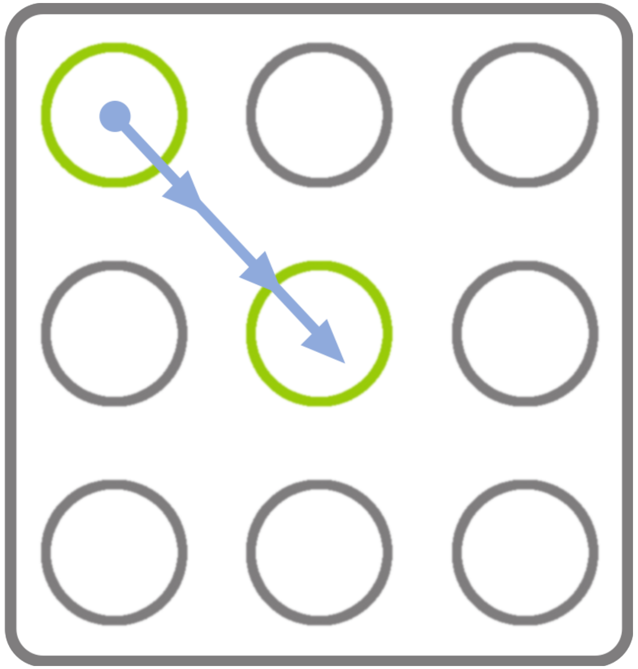
shake!

A. 패턴

출제자: 홍준표

- **수열**의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.

예제 4) [1, 5, 9, 6, 4]

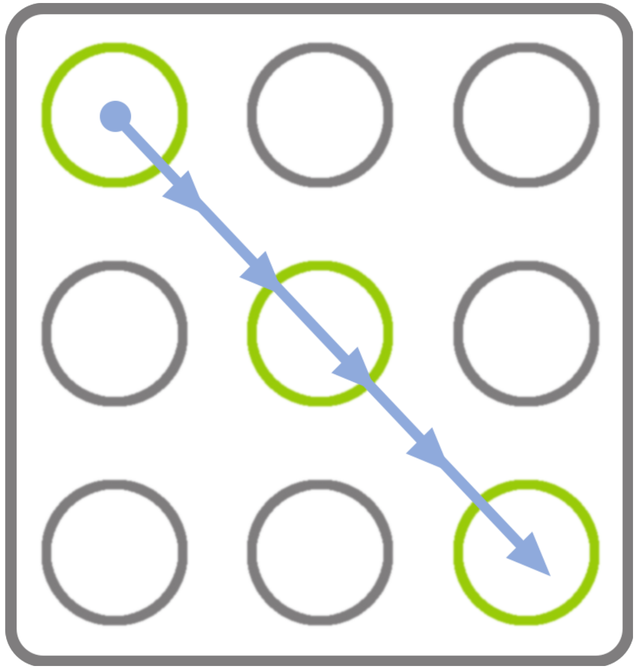


A. 패턴

출제자: 홍준표

- **수열**의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.

예제 4) [1, 5, 9, 6, 4]



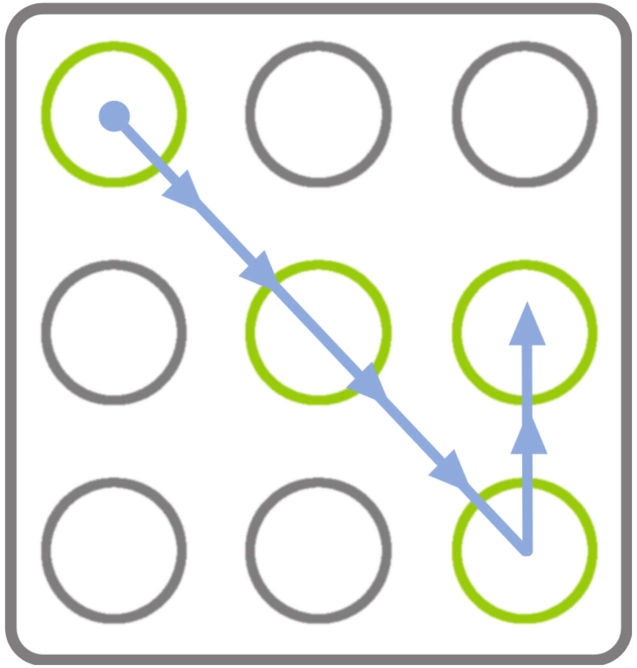
shake!

A. 패턴

출제자: 홍준표

- **수열**의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.

예제 4) [1, 5, **9, 6**, 4]



shake!

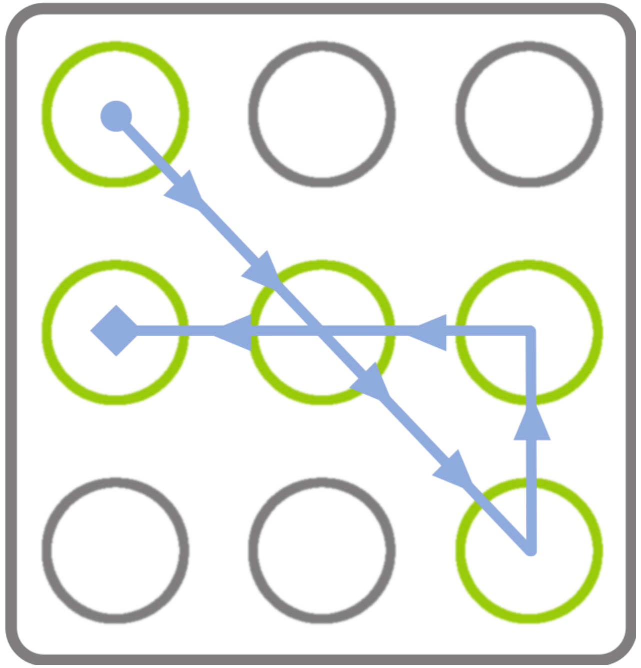
A. 패턴

출제자: 홍준표

- **수열**의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.

예제 4) [1, 5, 9, **6, 4**]

- 6-4 사이에 5번 점이 있지만, 이미 등장했기 때문에 가능한 패턴 입니다.



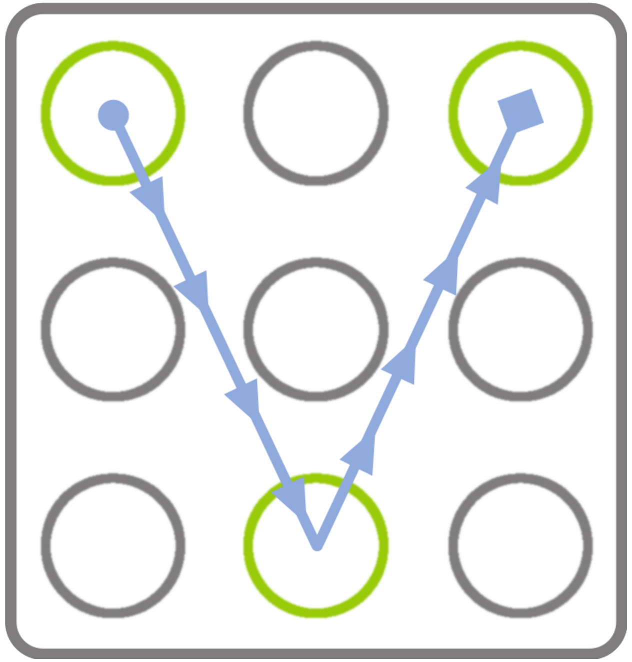
shake!

A. 패턴

출제자: 홍준표

- **수열**의 인접한 점을 연결해 만든 선분 위에는 아직 등장하지 않은 점이 있을 수 없다.

[1, 8, 4]

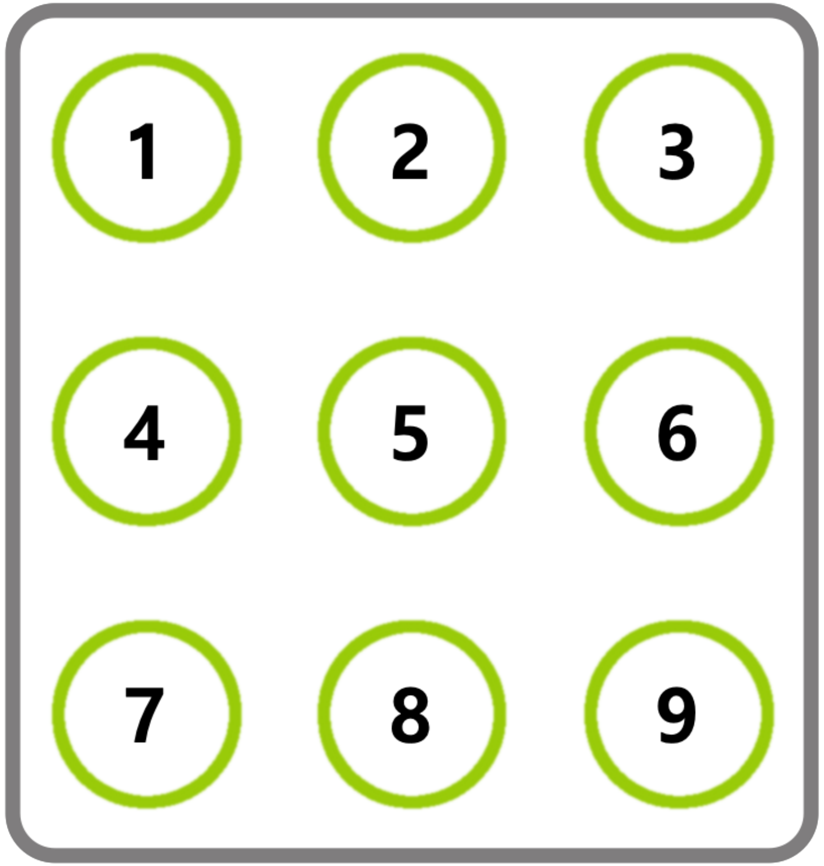


- 연결한 선분 위에 점이 없으므로 가능한 패턴 입니다.

A. 패턴

출제자: 홍준표

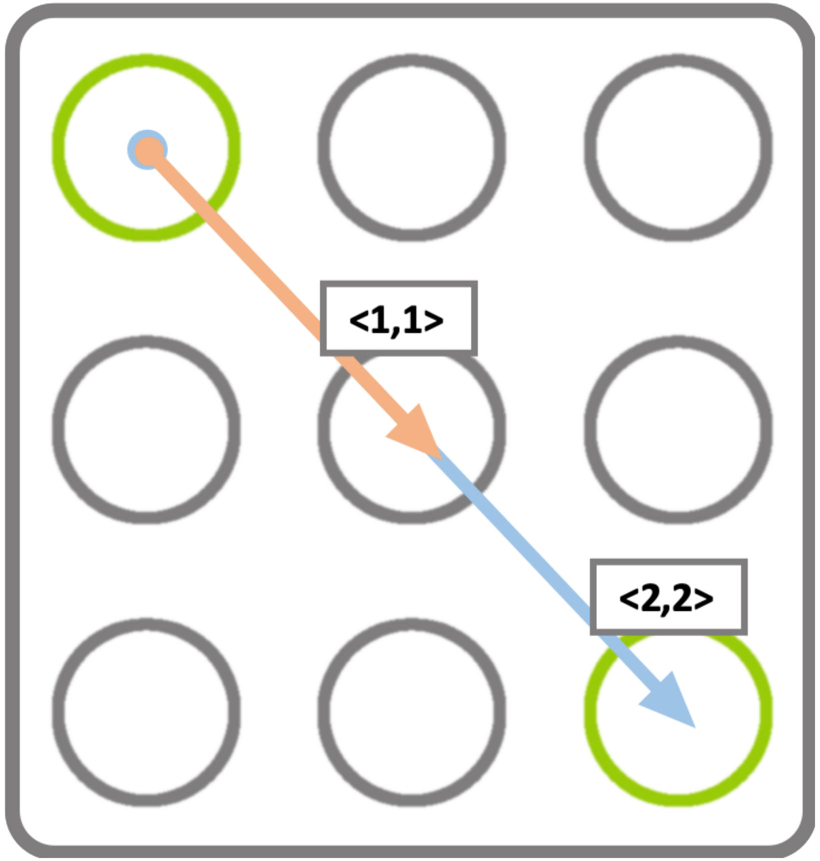
- 모든 참가자가 푸는것을 목표..
- 1-3 입력이 들어오면 2 체크,
  1-9 입력이 들어오면 5 체크,
  1-7 입력이 들어오면 4 체크 …
- 칸이 작아서 위와 같은 노가다로도 해결할 수 있다.



A. 패턴

출제자: 홍준표

- 만약 칸이 커진다면?
- a->b를 향하는 벡터의 '기약벡터'를 구해 a와 b 사이의 노드가 모두 체크 되어있는지 확인한다.
- 기약벡터는 최소공배수로 나누면 쉽게 구할 수 있다.

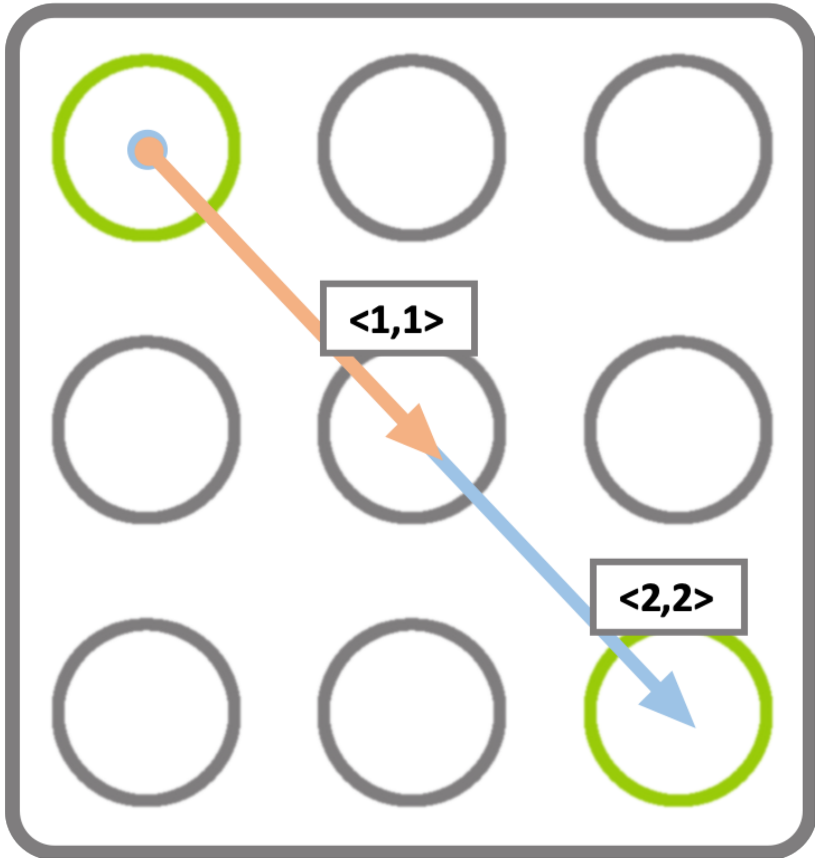


shake!

A. 패턴

출제자: 홍준표

- 시간복잡도?
- 정사각형 한 변의 길이를 *N* 이라고 할 때 패턴의 길이는 최대 *N*^2
- 점 사이에 최대 N개의 점이 들어올 수 있다
- 복잡도 : $O(N^3)$



shake!

F. 사탕 배달

정답자: ac: 26 / submission: 199

출제자: 구재현 (KAIST)

F. 사탕 배달

출제자： 구재현

- 3g의 사탕을 x개 먹고, 5g의 사탕을 y개 먹기 위해서는 $3x + 5y \leq w$ 를 만족해야 합니다.
- 근데, 달콤한 사탕이 있는데 굳이 안 먹을 필요는 없으니까...

shake!

F. 사탕 배달

출제자: 구재현

- 사탕이 있는데 굳이 안 먹을 필요는 없으니까...
- 3g의 사탕을 x개 먹었다면, 5g의 사탕은 $\left\lfloor \frac{w-3x}{5} \right\rfloor$개 먹으면 되겠네요.
- 물론 사탕을 -10개씩, 혹은 주어진 사탕보다 더 많이 먹으려고 시도하시면 WA를 받으실 겁니다.

F. 사탕 배달

출제자: 구재현

- 사탕의 개수 X를 고정했다면, 당도가 가장 큰 X개의 사탕을 먹으면 됩니다.
- 3g 사탕, 5g 사탕을 크기 역순으로 정렬하고, 모든 O(n) 개의 후보에 대해서 당도가 높은 사탕을 고르시면 됩니다.
- 정렬은 O(n log n)에 내장 함수를 사용하여 가능합니다.

F. 사탕 배달

출제자: 구재현

- 가능한 후보를 O(n) 시간에 판단하시면 O(n^2) 라 느립니다. O(1) 정도에 하셔야 합니다.
- SumOfBest3G[i] = SumOfBest3G[i-1] + (i번째로 당도가 높은 3g 사탕)
- .. 과 같은 배열을 만들어 놓는 식으로 해결하실 수 있습니다.
- 이 외에도 여러 방법이 있습니다. 위에서 설명한 방법이 가장 쉬운 방법은 아니지만, 가장 깔끔합니다.
- 아무튼 시간 복잡도는 O(n log n) 입니다.

shake!

H. 색깔 통일하기

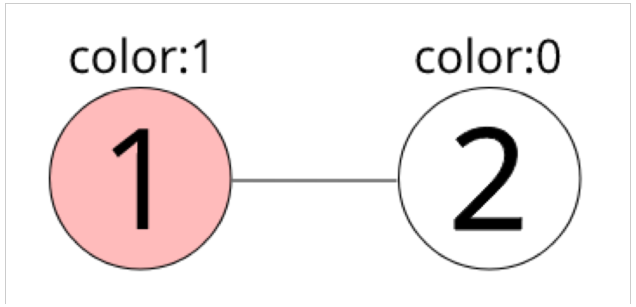
정답자: ac: 15 / submission: 52

출제자: 류호준 (성균관대학교)

H. 색깔 통일하기

출제자: 류호준

- 두 버튼만 있을때를 생각해보자.(N=2, C=4)

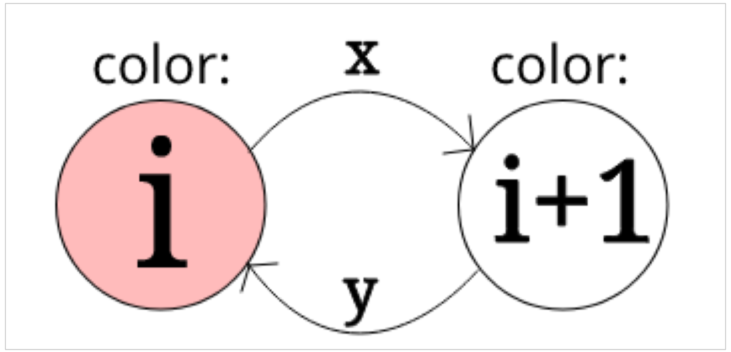


- 둘 모두 같은 색으로 만들기 위해
  1번은 3회 눌러야 하고
  2번은 1회 눌러야 한다.

H. 색깔 통일하기

출제자: 류호준

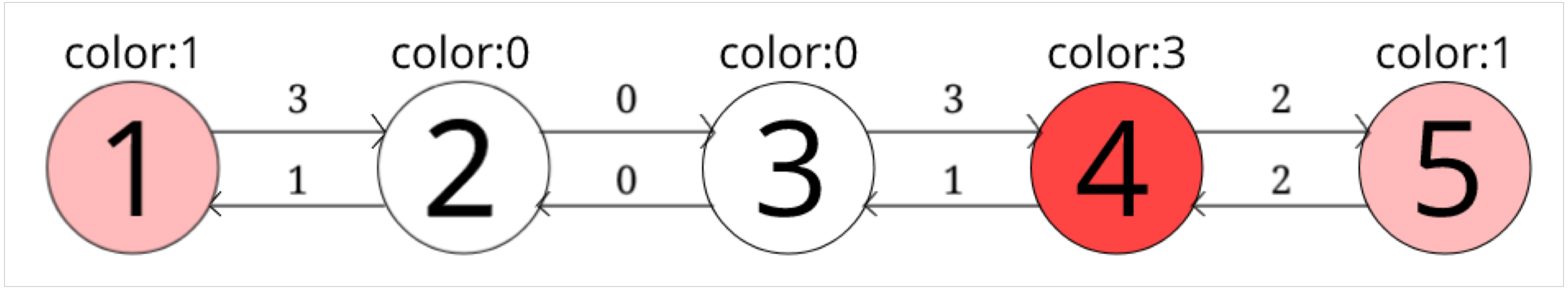
- i번 버튼을 x회 눌러야 i+1번 버튼과 색이 같아지고,
  i+1번 버튼을 y회 눌러야 i번 버튼과 색이 같아진다는 것을 다음과 같이 표현할 수 있다.



H. 색깔 통일하기

출제자: 류호준

- 이를 원래 예제 기준으로 다시 그리게 되면



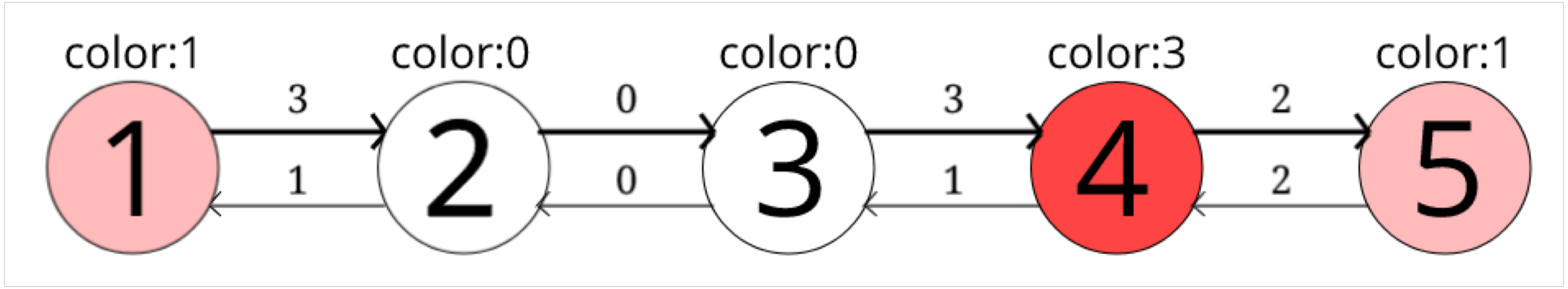
위와 같은 그래프가 나오게 된다.

- 연속한 같은 색 끼리는 한번에 변하니, 원래 문제는 각 정점에서부터 양 끝까지 도달하는 경로의 거리들 중 최소값을 구하는 문제가 된다.

H. 색깔 통일하기

출제자: 류호준

- 1번 버튼을 누른다 했을 때

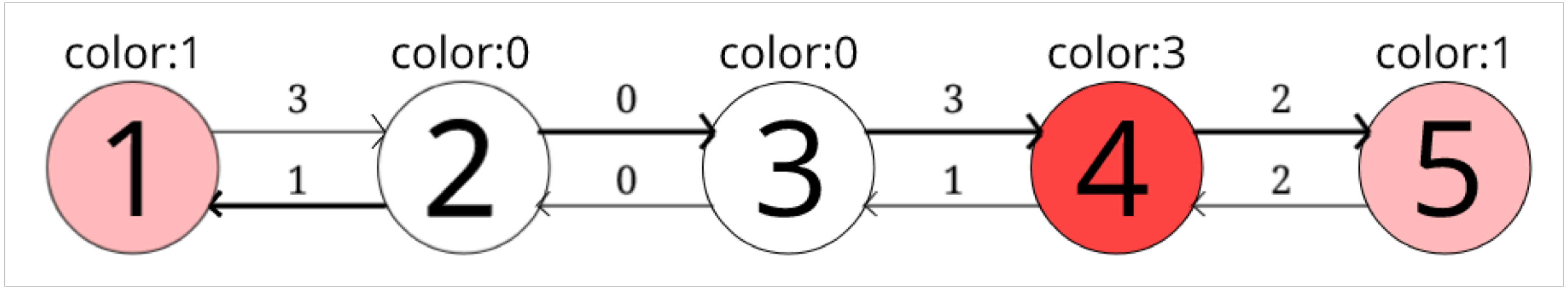


눌러야 할 횟수는 max(3+0+3+2 , 0) = 11이 된다.

H. 색깔 통일하기

출제자: 류호준

- 2번 버튼을 누른다 했을 때

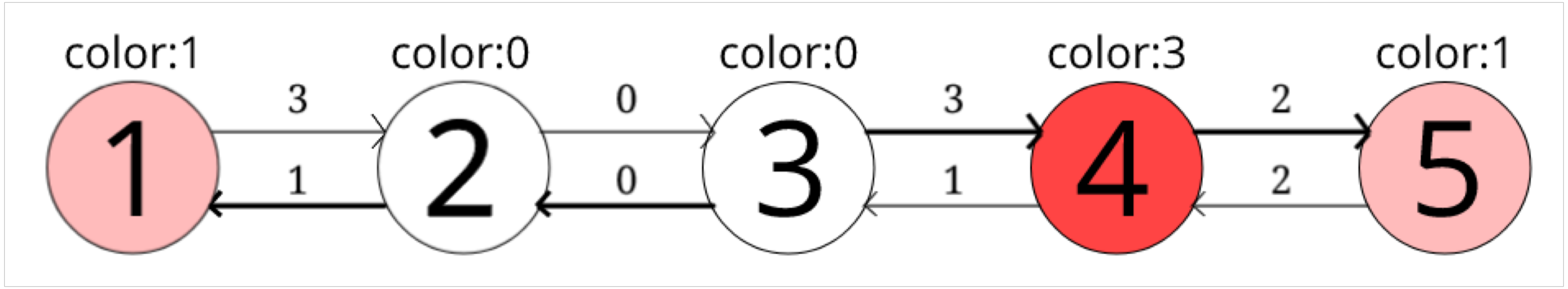


눌러야 할 횟수는 max(0+3+2 , 1) = 5가 된다.

H. 색깔 통일하기

출제자: 류호준

- 3번 버튼을 누른다 했을 때

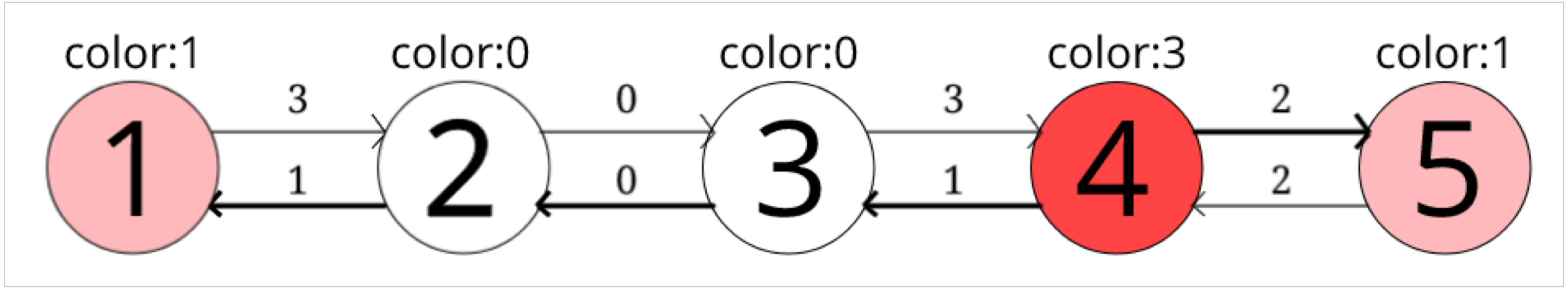


눌러야 할 횟수는 max(3+2 , 1+0) = 5가 된다.

H. 색깔 통일하기

출제자: 류호준

- 4번 버튼을 누른다 했을 때

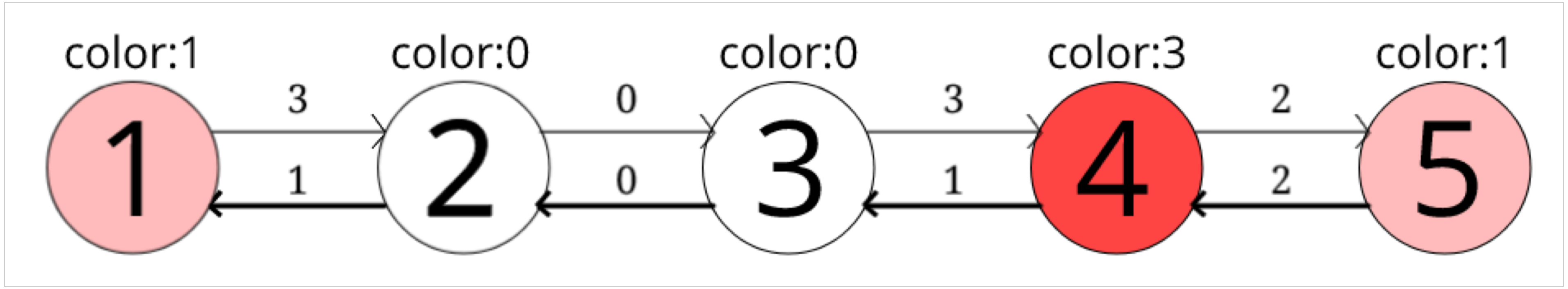


눌러야 할 횟수는 max(2 , 1+0+1) = 2가 된다.

H. 색깔 통일하기

출제자: 류호준

- 5번 버튼을 누른다 했을 때



눌러야 할 횟수는 max(0 , 1+0+1+2) = 4가 된다.

- 11, 5, 5, 2, 4 중 최소값인 2가 답이 된다.
- 반복문 돌면서 값 하나 빼주고, 하나 더해주는 방식으로 진행
- 시간복잡도: *O(N)*

shake!

B. NC 문자열

정답자: ac: 10 / submission: 87

출제자: 김현정 (아주대학교)

B. NC 문자열

출제자: 김현정

- NC 문자열을 찾지 말고, NC 문자열이 아닌 것을 찾아보자.
- NC 문자열이 아니라면?

$\underbrace{\text{...C...C...C}}_{\text{N이 없는 단어들}}$ ... $\underbrace{\text{N...N...N ...}}_{\text{C가 없는 단어들}}$

B. NC 문자열

출제자: 김현정

- N 다음에 C를 가진 단어를 NC 단어
- C 다음에 N을 가진 단어를 CN 단어
- C만 가진 단어들을 CC 단어
- N만 가진 단어들은 NN 단어라고 하자.

...C...C...C ... N...N...N ...

N이 없는 단어들   C가 없는 단어들

B. NC 문자열

출제자: 김현정

- CC 단어들은 앞에 위치해야 한다.
- NN 단어들은 뒤에 위치해야 한다.
- 그 경계에는 CN 단어가 1개 이하 들어갈 수 있다.
- NC 단어는 넣을 수 없다.

...C...C...C ... N...N...N ...

N이 없는 단어들 C가 없는 단어들

B. NC 문자열

출제자: 김현정

- 만들 수 있는 모든 문자열의 개수

  = nP0 + nP1 + nP2 + … nPn

- NC문자열이 아닌 문자열의 개수

  = (CN단어의 수 + 1) * (CC단어를 고르는 모든 경우)

  * (NN 단어를 고르는 모든 경우)

$$\underbrace{\ldots C \ldots C \ldots C}_{\text{N이 없는 단어들}} \;\ldots\; \underbrace{N \ldots N \ldots N \ldots}_{\text{C가 없는 단어들}}$$

shake!

B. NC 문자열

출제자: 김현정

- F(n) = nP0 + nP1 + nP2 + … nPn
  n개의 단어를 중복 없이 나열하는 모든 경우의 수

- F(n) = n * F(n – 1) + 1
- 이는 F(0) = 1 부터 시작하는 반복문으로 O(N)에 구할 수 있다.

- 답: F(N) - (CN문자열 수 + 1) * F(CC문자열 수) * F(NN문자열 수)

shake!

B. NC 문자열

출제자: 김현정

- 별해
- CC 문자열의 개수 = x / NN 문자열의 개수 = y 라고 할 때
- CC 문자열을 배치하는 모든 경우의 수: xP0 + xP1 + … + xPx
- NN 문자열을 배치하는 모든 경우의 수: yP0 + yP1 + … + yPy

- nPr을 상수항에 구할 수 있다면
  우리는 반복문을 통해 위 값을 O(N)에 구할 수 있다.

shake!

B. NC 문자열

출제자 : 김현정

- 별해
- Fac[i] = i! % mod 를 미리 구해두면
- nPr = Fac[n] / Fac[n – 1]

  = Fac[n] * (Fac[n – r]의 모듈러 역원) 으로 한 번에 구할 수 있다.

- 나눗셈을 곱셈으로 바꿀 수 있는 역원?

  => extended euclidean 알고리즘 or 페르마 소정리(소수)

C. 흰색으로 만들기

정답자: ac: 3 / submission: 9

출제자: ainta (서울대학교)

C. 흰색으로 만들기

출제자: 조승현

- Solution
- 항상 가능
  - 예제로 유추가능

shake!

C. 흰색으로 만들기

출제자: 조승현

- Solution
- 항상 가능
  - 예제로 유추가능
- 두 가지 연산만 있어도 충분
  - 인접한 칸들 전부 반전시키기
  - 자기자신 및 인접한 칸들 반전시키기
- Why?

C. 흰색으로 만들기

출제자: 조승현

- 처음은 모든 칸을 ‘2’로 설정 (인접한 칸 반전)

- 그 뒤에 검은색이 된 칸을 ‘3’으로 바꾸면..?

- 자기 자신의 색만 반전!

- 모든 칸이 흰색으로 바뀐다

shake!

E. 변호사들

정답자: ac: 2 / submission: 119

출제자: ainta (서울대학교)

E. 변호사들

출제자: 조승현

- Solution

- 그래프 G = (V, E)

- 각 변호사가 정점

- A가 B를 변호할 수 있으면 A에서 B로 가는 간선을 만든다

shake!

E. 변호사들

출제자: 조승현

- G의 간선 A -> B
  - 간선 B -> A가 없는 경우 : A가 B를 변호하는 것이 하지 않는 것보다 무조건 이득

- 역방향 간선이 없는 간선들은 모두 변호를 하는 것으로 결정한 후, G에는 양방향 간선만 남긴다.

shake!

E. 변호사들

출제자: 조승현

- 양방향 간선들로 이루어진 그래프 G
- 각 컴포넌트 별로 따로 생각

- 하나의 connected component에 이미 단방향 간선으로 변호를 받은 변호사가 있는 경우
  - 그 컴포넌트의 스패닝 트리 T
  - 변호를 받은 변호사를 루트로 하자
  - 부모노드가 자식노드를 변호하면 조건을 만족

E. 변호사들

출제자: 조승현

- 하나의 connected component에 이미 단방향 간선으로 변호를 받은 변호사가 없는 경우
  - 그 컴포넌트가 트리인 경우 – 정점 개수가 간선 개수보다 적으므로 조건을 만족하게 변호하는 것이 불가능
  - 트리가 아닌 경우 – 스패닝 트리를 잡은 후, 트리에 포함되지 않은 간선 하나를 잡아 이를 A<->B라 하면 A가 B를 변호한다고 할 수 있고, 이제 B를 루트로 하면 변호를 받은 변호사가 있는 경우와 동일함. 즉 조건을 만족하는 변호가 가능

E. 변호사들

출제자: 조승현

- 정리
  - 단방향 간선에 해당하는 변호는 하는 것으로 결정
  - 양방향 간선으로 이루어진 그래프 G
  - G에서 트리이면서 변호를 이미 받은 변호사가 하나도 없는 컴포넌트가 존재하면 NO
  - 그런 컴포넌트가 없으면 YES

shake!

G. 전쟁 중의 삶

정답자: ac: 3 / submission: 39

출제자: 구재현 (KAIST)

G. 전쟁 중의 삶

출제자: 구재현

- 무한 이진 트리에서 위험한 정점의 개수를 세는 문제입니다.
- 위험한 정점들은 트리 상에서 연결된 컴포넌트를 이룹니다.
- *(정확히는, 위험한 정점들을 포함하는 가장 작은 연결 컴포넌트를 이루게 됩니다! 이를 Steiner Tree라고 부릅니다.)*
- 연결 컴포넌트이고 트리이니, 정점의 개수는 간선보다 1 많습니다.

shake!

G. 전쟁 중의 삶

출제자: 구재현

- 위험한 **간선**의 개수를 세어 준 후 1을 더해서 출력해 봅시다.
- 위험한 **간선**이 있는 두 서브트리를 생각해 봅시다.
- 만약 양 서브트리에 모두 군부대가 있으면 해당 간선의 양 끝점은 모두 위험합니다. 고로 위험한 간선입니다.
- 그렇지 않다면 많아야 한 끝점이 위험하니, 위험한 간선이 아닙니다.

G. 전쟁 중의 삶

출제자: 구재현

- 위험한 정점보다 훨씬 깔끔하게 정의되네요. 이제 직접 구해봅시다.
- 위험한 간선은 완전 이진 트리의 부모와 자식을 잇고, 서브트리에 하나 이상의 군부대를 가집니다.
- 거꾸로, 모든 군부대에서 1번 정점으로 가는 경로를 전부 모으면, 위험한 간선의 후보를 모두 추릴 수 있습니다.

G. 전쟁 중의 삶

출제자: 구재현

- 여기서 중요한 관찰이 있는데, 바로 완전 이진 트리를 **트라이** 라고 생각할 수 있다는 것입니다.
- 각 군부대의 위치를 이진수로 표현하면, "100", "101", "110101 " 과 같은 이진 문자열이 됩니다.
- 맨 앞 1을 제거하고 이를 Trie에 넣으면, 모든 군부대에서 1번 정점으로 가는 경로를 저장할 수 있습니다.
- 이제 Trie에서 S로 표현되는 정점은, 실제 완전 이진 트리에서 1S의 이진수 값을 가지는 정점과 동일합니다.
- 노드 개수는 최대 50N개로 저장하기 충분합니다.

shake!

G. 전쟁 중의 삶

출제자: 구재현

- 위험한 간선은 자식 쪽 서브트리에도, 부모 쪽 서브트리에도 군부대가 있습니다.
- **위험한 간선은 자식 쪽 서브트리에 1 이상 N-1 이하의 군부대를 가집니다.**
- 트라이의 각 노드에 대해서, $DP[v]$ = (v의 서브트리에 몇 개의 노드가 있는가?) 를 Bottom-up DP로 계산합시다.
- $1 \leq DP[v] \leq N - 1$ 인 노드의 부모로 가는 간선은 위험합니다.
- 여기에 1을 더해서 출력해 주시면 AC!

shake!

D. 슈퍼브 다트

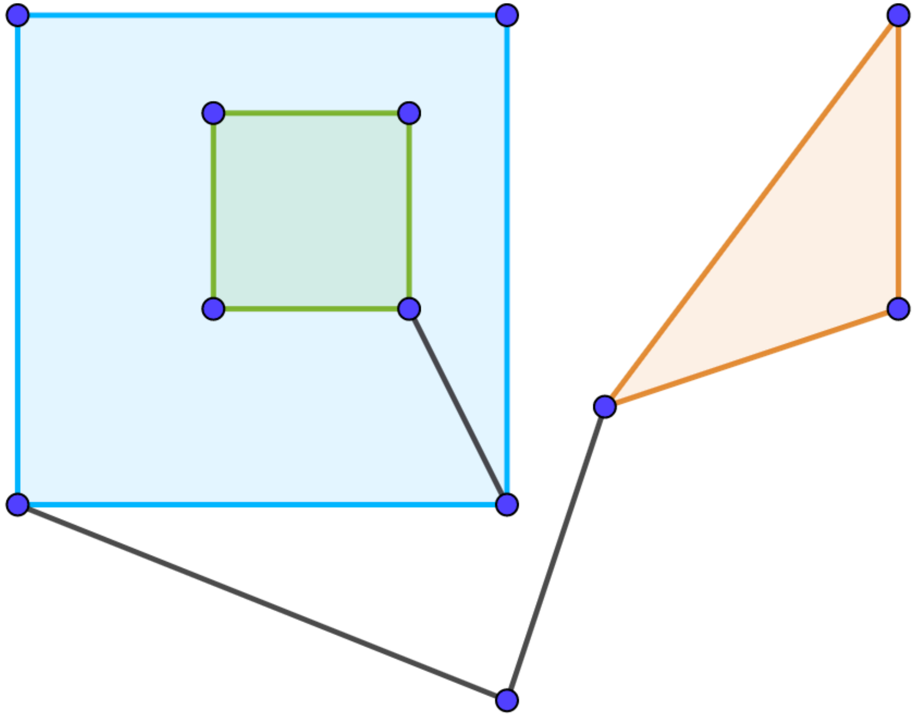
정답자: ac: 0 / submission: 0

출제자: 이종혁 (Superb AI)

D. 슈퍼브 다트

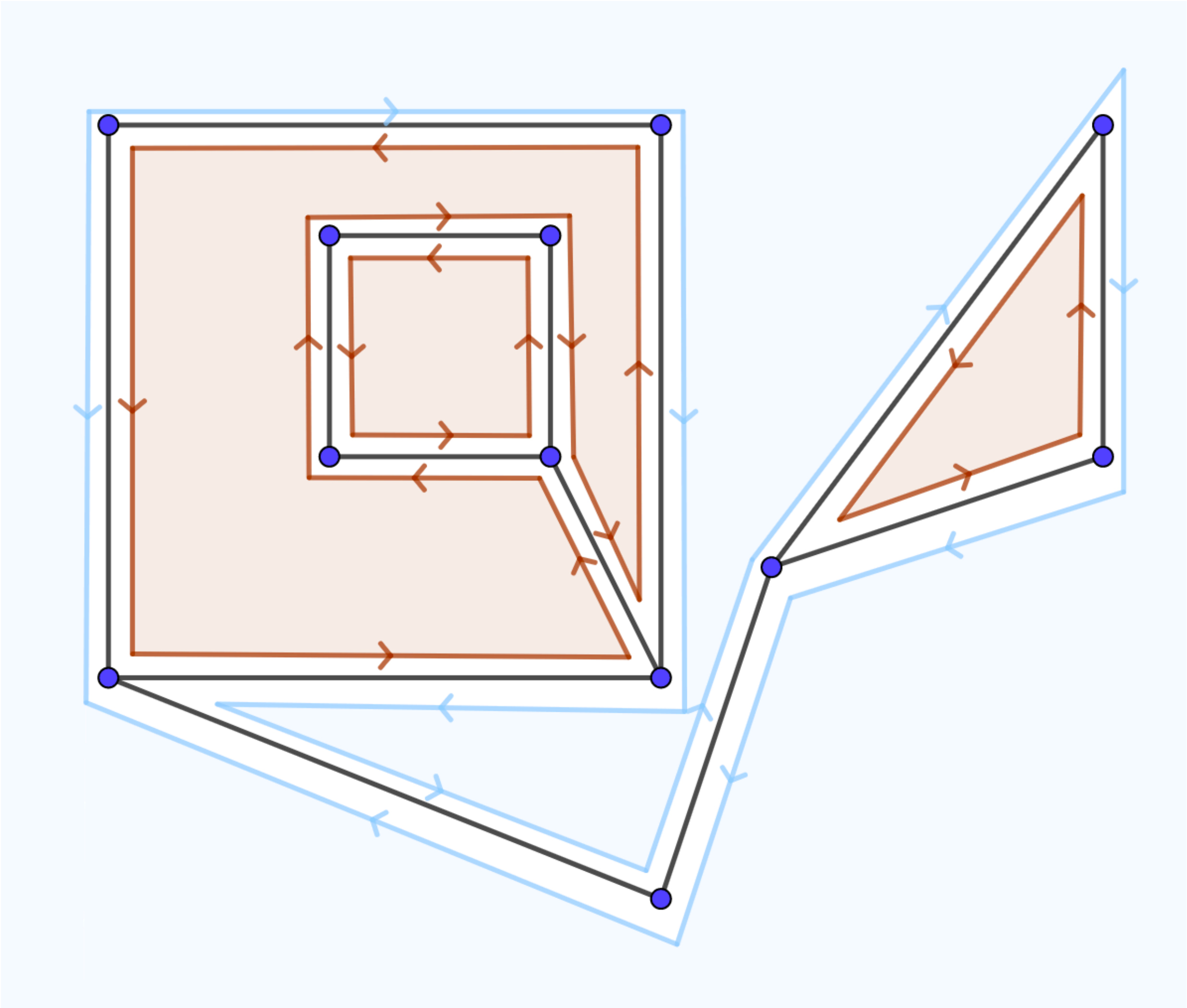
출제자 : 이종혁

- 평면그래프의 폴리곤을 찾는 문제
- 두 점을 연결하는 선분을 두갈래로 나눠보자
- i -> 2 x i, 2 x i + 1



D. 슈퍼브 다트

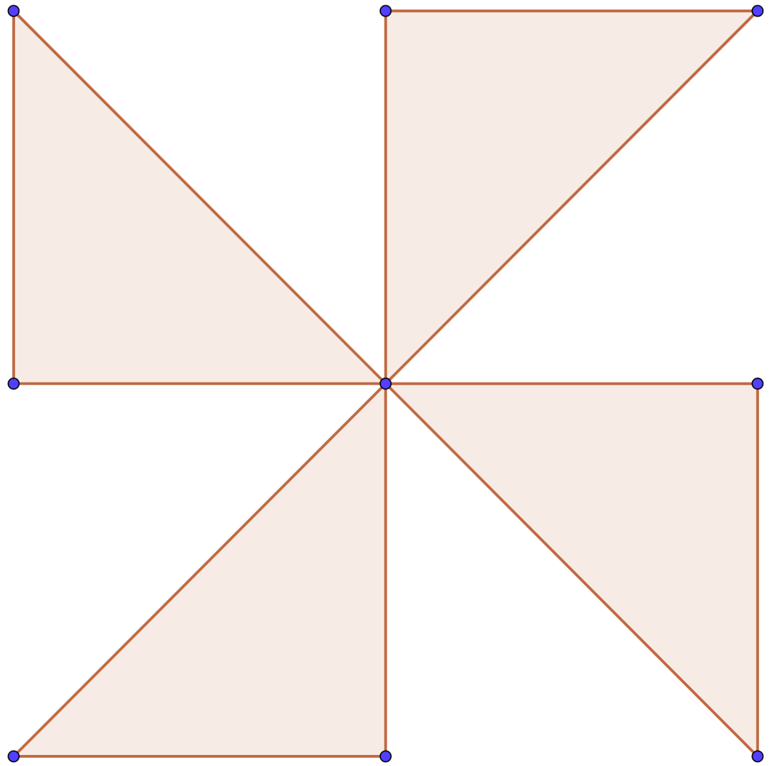
출제자: 이종혁

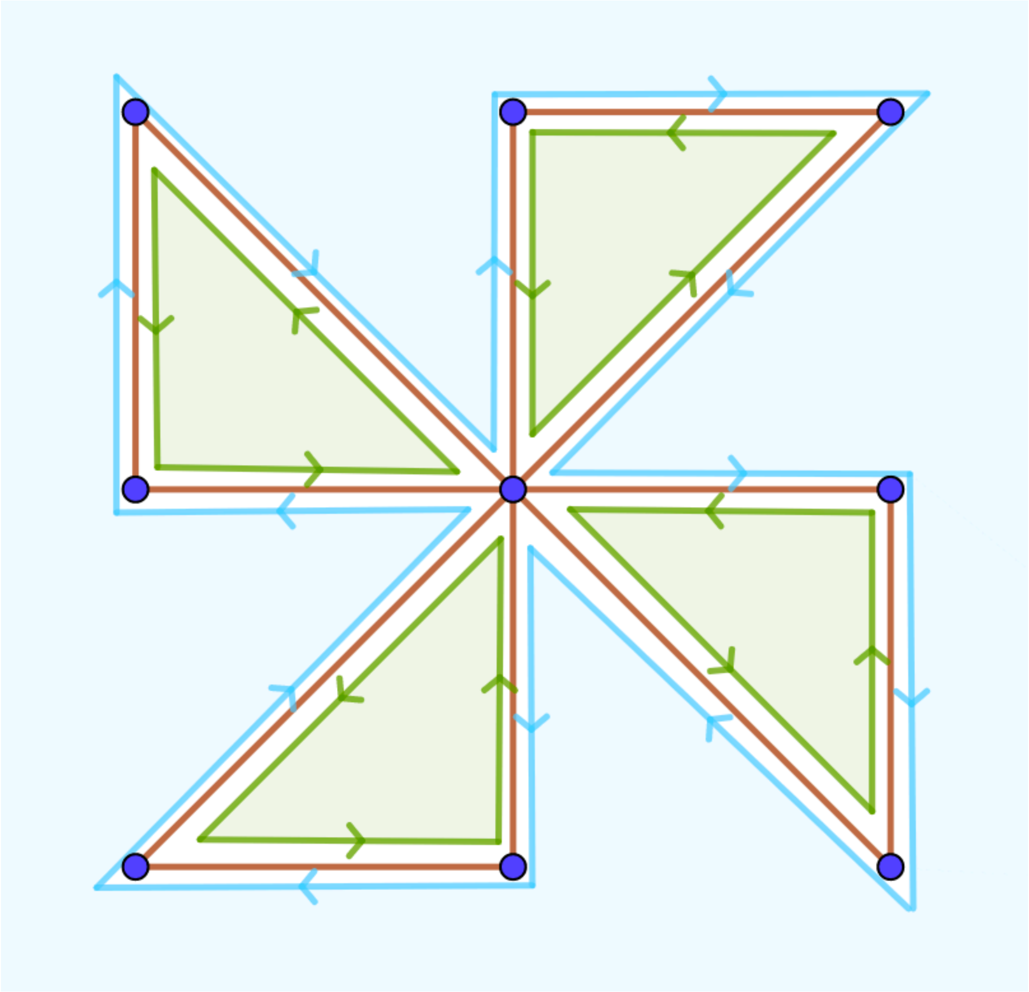


shake!

D. 슈퍼브 다트

출제자: 이종혁

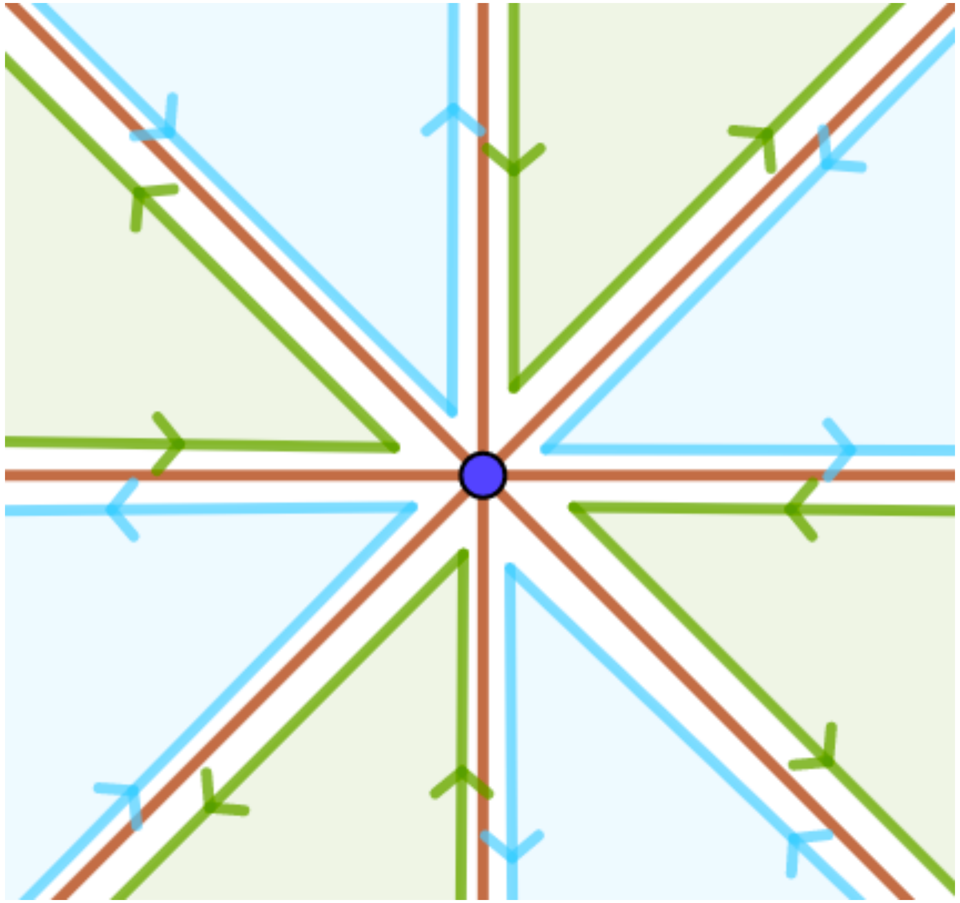




D. 슈퍼브 다트

출제자: 이종혁

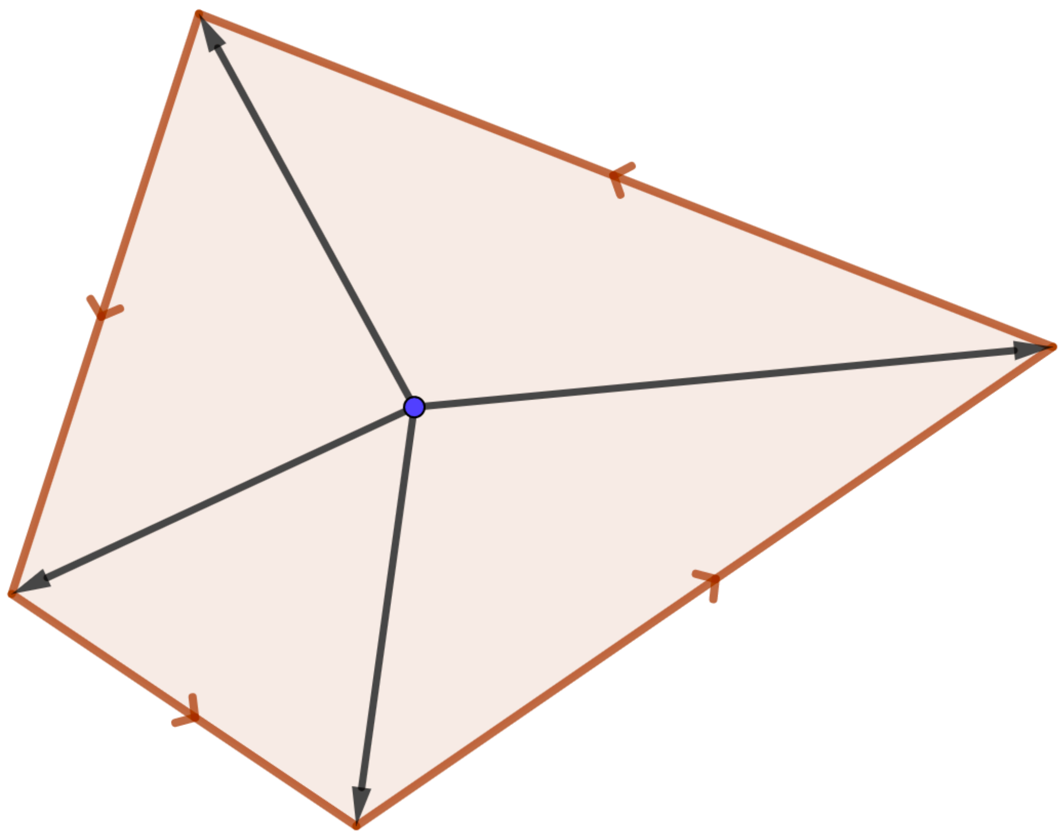
- 잘 관찰해보면..
- 인접한 간선들은 한 폴리곤의 모서리가 된다는 사실을 알 수 있다.
- 각 정점에 달린 간선을 시계방향으로 정렬 후 -> CCW
- 그림과 같이 폴리곤의 모서리가 되는 인접한 간선을 하나로 묶어준다.
  -> Disjoint Set!



D. 슈퍼브 다트

출제자: 이종혁

- 각 폴리곤의 넓이는 ?
- 임의의 점을 하나 고정한 뒤, 그림과 같이 다각형을 순회하며 얻는 외적값을 더한 후 반절로 나눠주면 다각형의 넓이
- merge 하는 과정에서 쉽게 해결할 수 있음



D. 슈퍼브 다트

출제자: 이종혁

- 폴리곤의 넓이가 음수라면 ?
  - 바깥쪽을 의미하므로 무시한다.
- 각 간선을 각도순 정렬 + 인접한 간선 연결 + 넓이 정렬
- 폴리곤의 최대개수는 E에 비례한다.

(삼각형의 추가를 상상해보자)

- 시간복잡도 : $O(\mathrm{E} \log E)$