ICPC Asia Seoul Regional Contest 2024

Solutions Presentation

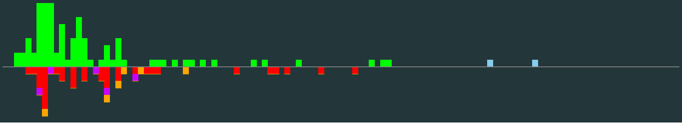
November 23, 2024

icpc.foundation

icpc Asia Seoul

A: Bottles

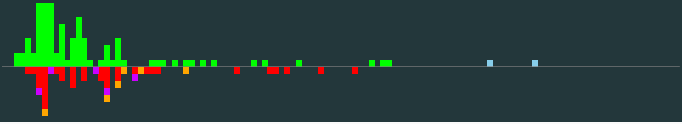
Proposer: 이상헌, Setter: 이인복



문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

A: Bottles

Proposer: 이상헌, Setter: 이인복



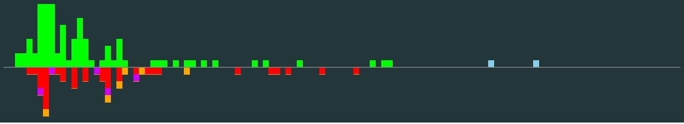
문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

풀이:

- 선수 i가 구간 j를 통과하는데 걸리는 시간 $=: a_{i,j}$.

A: Bottles

Proposer: 이상헌, Setter: 이인복



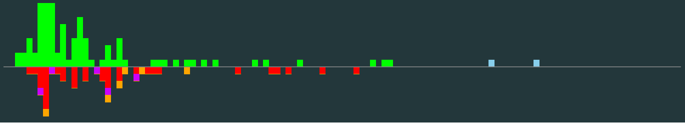
문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

풀이:

- 선수 i가 구간 j를 통과하는데 걸리는 시간 $=: a_{i,j}$.
- $b_{i,j} := a_{i,1} + a_{i,2} + \cdots + a_{i,j}$.

A: Bottles

Proposer: 이상헌, Setter: 이인복



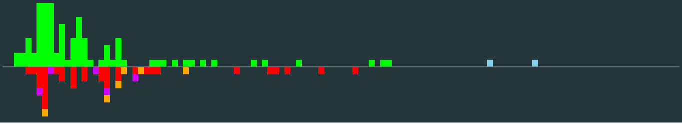
문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

풀이:

- 선수 i가 구간 j를 통과하는데 걸리는 시간 $=: a_{i,j}$.
- $b_{i,j} := a_{i,1} + a_{i,2} + \cdots + a_{i,j}$.
- 구간 j에 선수 i가 머무르는 시간대 $= (b_{i,j-1}, b_{i,j})$.

A: Bottles

Proposer: 이상헌, Setter: 이인복



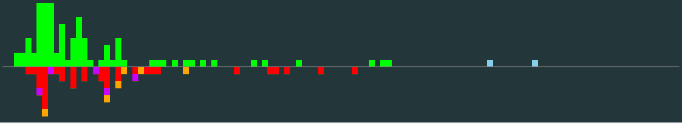
문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

풀이:

- 선수 i가 구간 j를 통과하는데 걸리는 시간 $=: a_{i,j}$.
- $b_{i,j} := a_{i,1} + a_{i,2} + \cdots + a_{i,j}$.
- 구간 j에 선수 i가 머무르는 시간대 $= \left(b_{i,j-1}, b_{i,j}\right)$.
- 구간 j의 선수의 수는
  - 시각 $b_{*,j-1}$일 때 1 증가.
  - 시각 $b_{*,j}$일 때 1 감소.

A: Bottles

Proposer: 이상헌, Setter: 이인복



문제: n 명의 선수가 m 개의 구간을 통과하는데 걸리는 시간이 주어질 때, 각 구간에 대해 **가장 선수가 많을 때의 사람 수**를 구하라.

풀이:

- 선수 i가 구간 j를 통과하는데 걸리는 시간 $=: a_{i,j}$.
- $b_{i,j} := a_{i,1} + a_{i,2} + \cdots + a_{i,j}$.
- 구간 j에 선수 i가 머무르는 시간대 $= \left(b_{i,j-1}, b_{i,j}\right)$.
- 구간 j의 선수의 수는
  - 시각 $b_{*,j-1}$일 때 1 증가.
  - 시각 $b_{*,j}$일 때 1 감소.

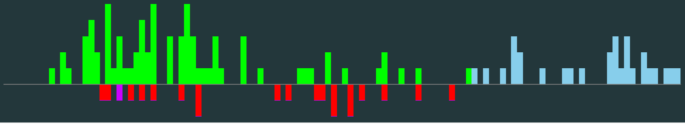
유의: 시각이 같을 때의 처리에 유의해야 합니다.

복잡도: $\mathcal{O}(nm \lg n)$.

통계량: 총 제출 135개, 정답 86개, 미공개 2개.

L: Triangle

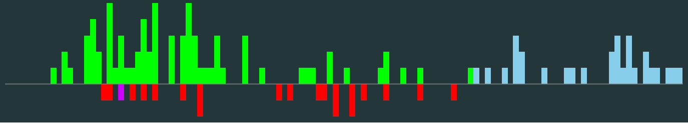
Proposer: 허성우, Setter: 허성우



문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

L: Triangle

Proposer: 허성우, Setter: 허성우



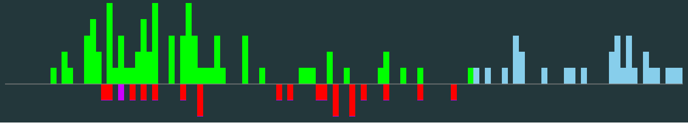
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은,

L: Triangle

Proposer: 허성우, Setter: 허성우



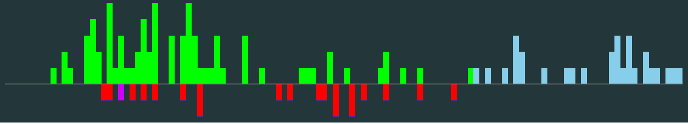
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$,

L: Triangle

Proposer: 허성우, Setter: 허성우



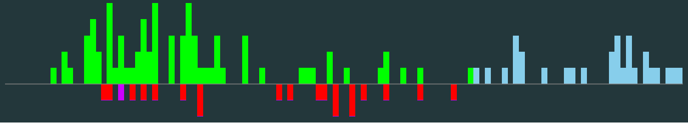
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때,

L: Triangle

Proposer: 허성우, Setter: 허성우



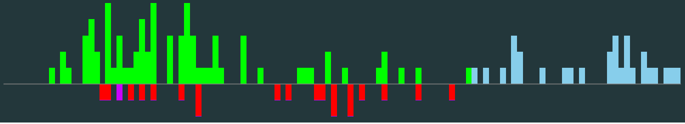
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때, $\left(p_x + n \times \frac{q_x - p_x}{g}, p_y + n \times \frac{q_y - p_y}{g}\right)$.

L: Triangle

Proposer: 허성우, Setter: 허성우



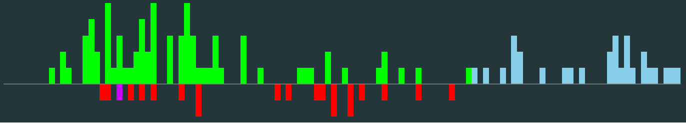
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때, $\left(p_x + n \times \frac{q_x - p_x}{g}, p_y + n \times \frac{q_y - p_y}{g}\right)$.
- 각 선분에서 잡은 정수점의 n 값을 각각 i, j, k라고 하자.

L: Triangle

Proposer: 허성우, Setter: 허성우



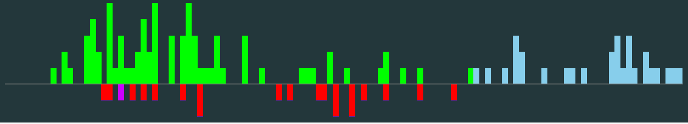
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 $(p_x, p_y), (q_x, q_y)$를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때, $\left(p_x + n \times \frac{q_x - p_x}{g}, p_y + n \times \frac{q_y - p_y}{g}\right)$.
- 각 선분에서 잡은 정수점의 n 값을 각각 i, j, k라고 하자.
- 세 점의 x, y 좌표는 각각 i, j, k에 대한 일차식.

L: Triangle

Proposer: 허성우, Setter: 허성우



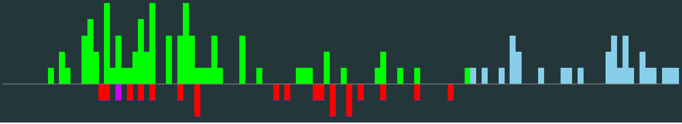
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 (p_x, p_y), (q_x, q_y)를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때, $\left(p_x + n \times \frac{q_x - p_x}{g}, p_y + n \times \frac{q_y - p_y}{g}\right)$.
- 각 선분에서 잡은 정수점의 n 값을 각각 i, j, k라고 하자.
- 세 점의 x, y 좌표는 각각 i, j, k에 대한 일차식.
- 새로 만든 삼각형의 넓이는 ij, jk, ki에 대한 일차식.

L: Triangle

Proposer: 허성우, Setter: 허성우



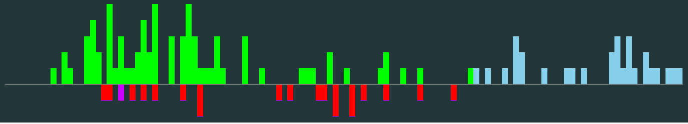
문제: 삼각형 $\triangle ABC$가 주어진다. 각 선분에서 꼭짓점이 아닌 정수점을 잡아 만들 수 있는 삼각형 중 **넓이의 최댓값과 최솟값**을 구하라.

풀이:

- 두 점 $(p_x, p_y), (q_x, q_y)$를 잇는 선분 위의 정수점은, $g := \gcd(|p_x - q_x|, |p_y - q_y|)$, $1 \le n < g$일 때, $\left(p_x + n \times \frac{q_x - p_x}{g}, p_y + n \times \frac{q_y - p_y}{g}\right)$.
- 각 선분에서 잡은 정수점의 n 값을 각각 i, j, k라고 하자.
- 세 점의 x, y 좌표는 각각 i, j, k에 대한 일차식.
- 새로 만든 삼각형의 넓이는 ij, jk, ki에 대한 일차식.
- 고로, i, j, k가 각각의 범위 내에서 최대/최소를 가질 때 중에서, 넓이가 최대 혹은 최소가 되는 순간이 반드시 존재.

L: Triangle

Proposer: 허성우, Setter: 허성우

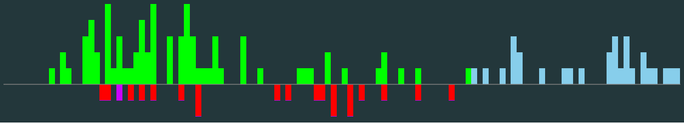


풀이:

- 2^3 개의 삼각형을 모두 시도.
- 삼각형 넓이 계산은 신발끈 공식으로.

L: Triangle

Proposer: 허성우, Setter: 허성우



풀이:

- 2^3 개의 삼각형을 모두 시도.
- 삼각형 넓이 계산은 신발끈 공식으로.

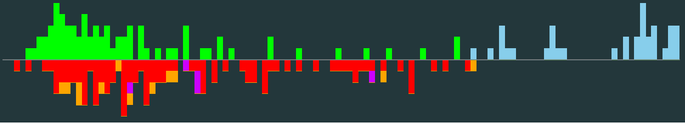
복잡도: $\mathcal{O}\left(2^3 + 3\lg(2 \times 10^9)\right)$.

번외: 볼록 n각형이 주어진다면?

통계량: 총 제출 124개, 정답 74개, 미공개 29개.

B: Cards Flipping

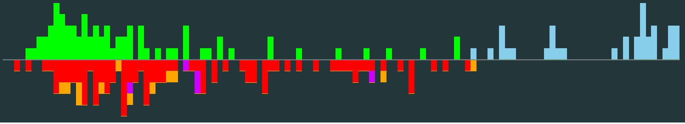
Proposer: 정현우, Setter: 정현우



문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



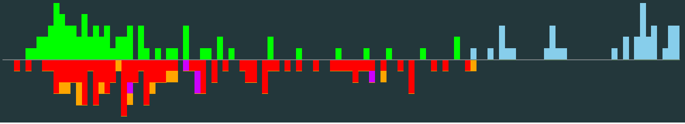
문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

- 각 정수를 하나의 정점으로 생각.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



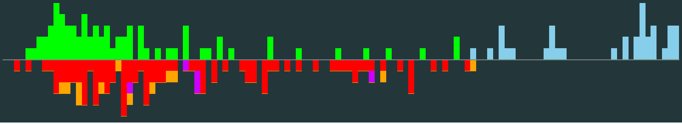
문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



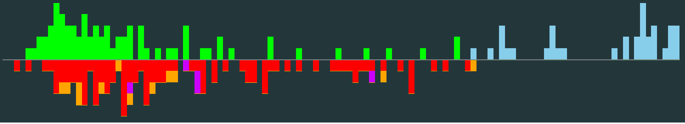
문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.
- 각 간선마다 양끝 정점 중 하나를 고를 수 있을 때, 선택된 정점의 수를 최대화.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

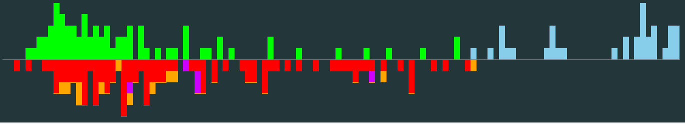
- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.
- 각 간선마다 양끝 정점 중 하나를 고를 수 있을 때, 선택된 정점의 수를 최대화.

풀이:

- 연결 컴포넌트별로 독립적으로 해결.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

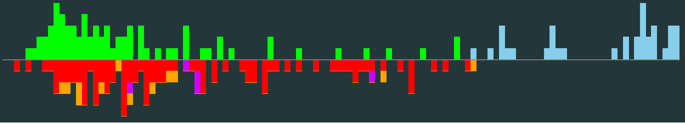
- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.
- 각 간선마다 양끝 정점 중 하나를 고를 수 있을 때, 선택된 정점의 수를 최대화.

풀이:

- 연결 컴포넌트별로 독립적으로 해결.
- 컴포넌트가 트리인 경우, 하나의 정점을 제외한 모든 정점을 선택 가능.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



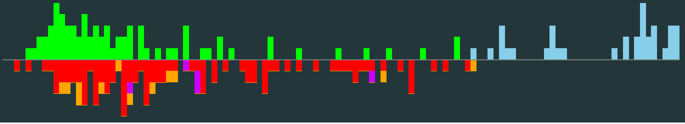
문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:
- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.
- 각 간선마다 양끝 정점 중 하나를 고를 수 있을 때, 선택된 정점의 수를 최대화.

풀이:
- 연결 컴포넌트별로 독립적으로 해결.
- 컴포넌트가 트리인 경우, 하나의 정점을 제외한 모든 정점을 선택 가능.
- 그 외의 경우, 모든 정점을 선택 가능.

B: Cards Flipping

Proposer: 정현우, Setter: 정현우



문제: 양면에 정수가 적혀있는 n 장의 카드가 놓여 있다. 카드를 적절히 뒤집어, **앞면에 적힌 서로 다른 정수들의 수를 최대화**하라.

관찰:

- 각 정수를 하나의 정점으로 생각.
- 각 카드의 양면에 적힌 두 정점을 간선으로 연결.
- 각 간선마다 양끝 정점 중 하나를 고를 수 있을 때, 선택된 정점의 수를 최대화.

풀이:

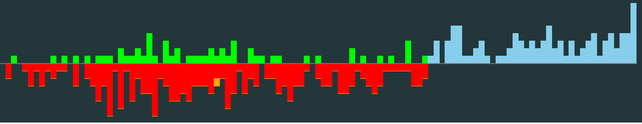
- 연결 컴포넌트별로 독립적으로 해결.
- 컴포넌트가 트리인 경우, 하나의 정점을 제외한 모든 정점을 선택 가능.
- 그 외의 경우, 모든 정점을 선택 가능.

복잡도: $\mathcal{O}(n)$.

통계량: 총 제출 215개, 정답 72개, 미공개 35개.

J: Street Development

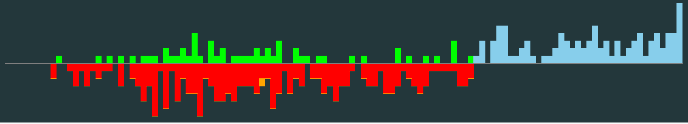
Proposer: **김재훈**, Setter: **조승범**



문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

J: Street Development

Proposer: 김재훈, Setter: 조승범



문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.

J: Street Development

Proposer: 김재훈, Setter: 조승범

문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.
- 어떤 i가 존재하여, 항상 다음과 같은 해 구성이 가능.

J: Street Development

Proposer: 김재훈, Setter: 조승범

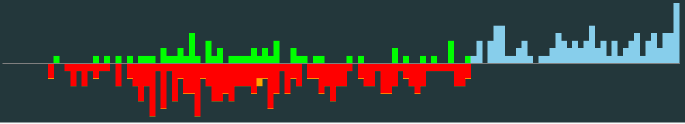
문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.
- 어떤 i가 존재하여, 항상 다음과 같은 해 구성이 가능.
  - 로봇 1이 오른쪽으로 w 이동.

J: Street Development

Proposer: 김재훈, Setter: 조승범



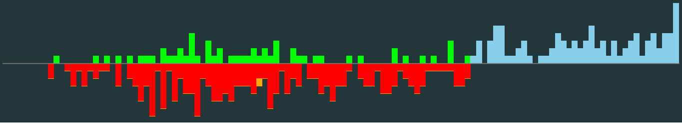
문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.
- 어떤 i가 존재하여, 항상 다음과 같은 해 구성이 가능.
  - 로봇 1이 오른쪽으로 w 이동.
  - 로봇 2가 로봇 1과 정보 공유를 위해 왼쪽으로 이동한 후, 남은 거리만큼 오른쪽으로 이동.

J: Street Development

Proposer: 김재훈, Setter: 조승범



문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.
- 어떤 i가 존재하여, 항상 다음과 같은 해 구성이 가능.
  - 로봇 1이 오른쪽으로 w 이동.
  - 로봇 2가 로봇 1과 정보 공유를 위해 왼쪽으로 이동한 후, 남은 거리만큼 오른쪽으로 이동.
  - $\vdots$
  - 로봇 i가 로봇 $i-1$과 정보 공유를 위해 왼쪽으로 이동한 후, 남은 거리만큼 오른쪽으로 이동.

J: Street Development

Proposer: 김재훈, Setter: 조승범

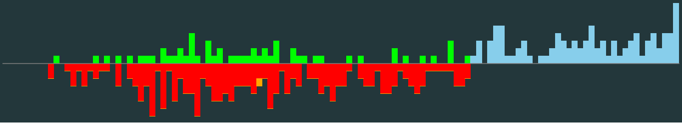
문제: 구간 $[0, L]$ 위에 n 개의 로봇이 놓여있다. 모든 로봇의 정보를 알고 있는 로봇이 존재하기 위해 **단일 로봇이 움직여야 하는 총 거리의 최솟값**을 구하라.

풀이:

- 답이 w 이하인지 판정하자.
- 어떤 i가 존재하여, 항상 다음과 같은 해 구성이 가능.
  - 로봇 1이 오른쪽으로 w 이동.
  - 로봇 2가 로봇 1과 정보 공유를 위해 왼쪽으로 이동한 후, 남은 거리만큼 오른쪽으로 이동.
  - $\vdots$
  - 로봇 i가 로봇 $i-1$과 정보 공유를 위해 왼쪽으로 이동한 후, 남은 거리만큼 오른쪽으로 이동.
  - 로봇 $n, n-1, \cdots, i+1$에 대해서는 대칭적으로 진행.

J: Street Development

Proposer: 김재훈, Setter: 조승범

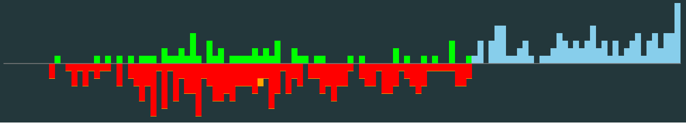


풀이:

- 앞에서 소개한 전략대로 이동할 때, 로봇 i와 $i+1$의 최종 위치를 각각 R_i, L_{i+1}라고 하자.

J: Street Development

Proposer: 김재훈, Setter: 조승범

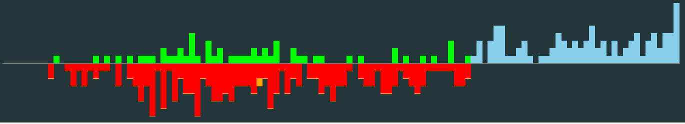


풀이:

- 앞에서 소개한 전략대로 이동할 때, 로봇 i와 $i+1$의 최종 위치를 각각 R_i, L_{i+1}라고 하자.
- R_j, L_j의 값은 각각 R_{j-1}, L_{j+1}로부터 계산 가능.

J: Street Development

Proposer: 김재훈, Setter: 조승범

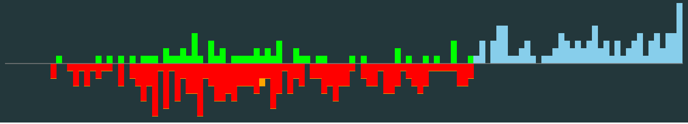


풀이:

- 앞에서 소개한 전략대로 이동할 때, 로봇 i와 $i+1$의 최종 위치를 각각 R_i, L_{i+1}라고 하자.
- R_j, L_j의 값은 각각 R_{j-1}, L_{j+1}로부터 계산 가능.
- $R_i \geq L_{i+1}$인 i가 존재하면 가능.

J: Street Development

Proposer: 김재훈, Setter: 조승범



풀이:

- 앞에서 소개한 전략대로 이동할 때, 로봇 i와 $i+1$의 최종 위치를 각각 R_i, L_{i+1}라고 하자.
- R_j, L_j의 값은 각각 R_{j-1}, L_{j+1}로부터 계산 가능.
- $R_i \geq L_{i+1}$인 i가 존재하면 가능.

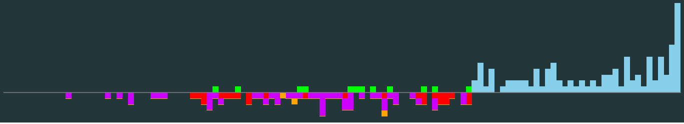
유의: w가 너무 작아, 두 인접한 로봇이 아예 만나지 못하는 경우를 잘 판정해야 합니다.

복잡도: $\mathcal{O}(n \lg L)$.

통계량: 총 제출 351개, 정답 56개, 미공개 96개.

C: Colorful Quadrants

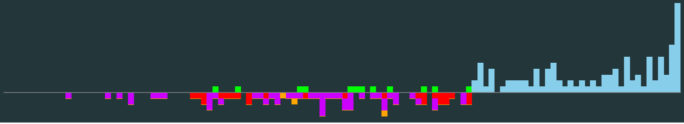
Proposer: 윤상덕, Setter: 윤상덕



문제: $n \times n$ 격자 중 일부에 정수가 적혀있다. 빈 격자점을 기준으로 네 사분면 각각에서 정수를 하나씩 뽑아 서로 다르게 만들 수 있는, **다채로운 사분면을 갖는 격자점의 수**를 세어라.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕



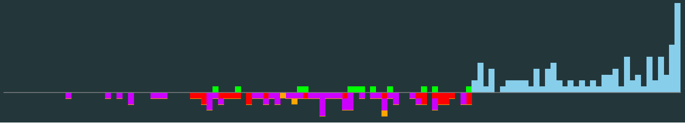
문제: $n \times n$ 격자 중 일부에 정수가 적혀있다. 빈 격자점을 기준으로 네 사분면 각각에서 정수를 하나씩 뽑아 서로 다르게 만들 수 있는, **다채로운 사분면을 갖는 격자점의 수**를 세어라.

관찰:

- 어떤 격자점이 다채로운 사분면을 갖는지 판정하기 위해서는, 각 사분면에 속한 **최대 네 종류의 정수**만 알면 충분.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕



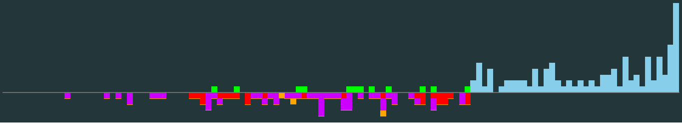
문제: $n \times n$ 격자 중 일부에 정수가 적혀있다. 빈 격자점을 기준으로 네 사분면 각각에서 정수를 하나씩 뽑아 서로 다르게 만들 수 있는, **다채로운 사분면을 갖는 격자점의 수**를 세어라.

관찰:

- 어떤 격자점이 다채로운 사분면을 갖는지 판정하기 위해서는, 각 사분면에 속한 **최대 네 종류의 정수**만 알면 충분.
  - 이후에는 4중 반복문 혹은 이분 매칭으로 빠르게 판정 가능.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕



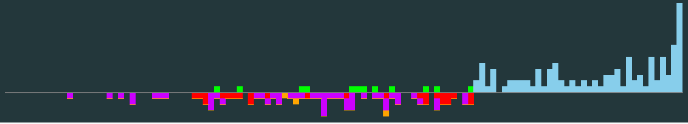
문제: $n \times n$ 격자 중 일부에 정수가 적혀있다. 빈 격자점을 기준으로 네 사분면 각각에서 정수를 하나씩 뽑아 서로 다르게 만들 수 있는, **다채로운 사분면을 갖는 격자점의 수**를 세어라.

관찰:

- 어떤 격자점이 다채로운 사분면을 갖는지 판정하기 위해서는, 각 사분면에 속한 **최대 네 종류의 정수**만 알면 충분.
  - 이후에는 4중 반복문 혹은 이분 매칭으로 빠르게 판정 가능.
- 격자점 (i, j)의 좌상단 사분면에 속한 정수들의 집합 $=: UL_{i,j}$
- $UL_{i,j}$에서 원소 네 개를 빠르게 알아내야 한다.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕

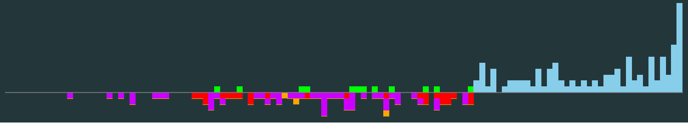


관찰:

- $[1, i-1]$ 번째 행에 놓인 정수들에 대해 (가장 작은 열 번호, 그 정수의 값)을 정렬.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕

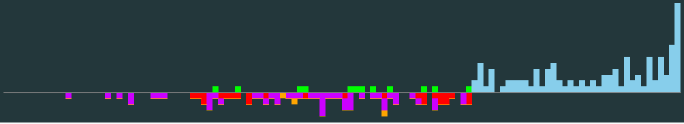


관찰:

- $[1, i-1]$ 번째 행에 놓인 정수들에 대해 (가장 작은 열 번호, 그 정수의 값)을 정렬.
- 가장 작은 네 순서쌍만으로도 $UL_{i,1}, UL_{i,2}, \cdots, UL_{i,n}$을 모두 결정 가능.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕

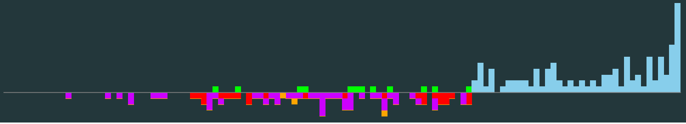


관찰:

- $[1, i-1]$ 번째 행에 놓인 정수들에 대해 (가장 작은 열 번호, 그 정수의 값)을 정렬.
- 가장 작은 네 순서쌍만으로도 $UL_{i,1}, UL_{i,2}, \cdots, UL_{i,n}$을 모두 결정 가능.
- i를 증가하면서 모든 $UL_{*,*}$를 $\mathcal{O}(4n(n+k))$에 계산 가능.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕



관찰:

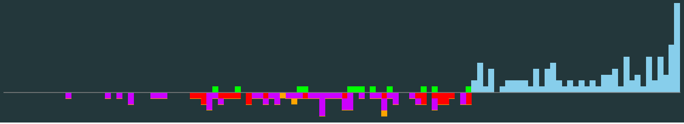
- $[1, i-1]$ 번째 행에 놓인 정수들에 대해 (가장 작은 열 번호, 그 정수의 값)을 정렬.
- 가장 작은 네 순서쌍만으로도 $UL_{i,1}, UL_{i,2}, \cdots, UL_{i,n}$ 을 모두 결정 가능.
- i를 증가하면서 모든 $UL_{*,*}$ 를 $\mathcal{O}(4n(n+k))$에 계산 가능.

풀이:

- 네 사분면 각각에 대해 집합 $UL_{*,*}$, $UR_{*,*}$,, $LL_{*,*}$, $LR_{*,*}$ 을 모두 계산.
- 각 빈 격자점에 대해, 네 집합을 사용해서 다채로운 사분면을 갖는지 판정.

C: Colorful Quadrants

Proposer: 윤상덕, Setter: 윤상덕



관찰:

- $[1, i-1]$ 번째 행에 놓인 정수들에 대해 (가장 작은 열 번호, 그 정수의 값)을 정렬.
- 가장 작은 네 순서쌍만으로도 $UL_{i,1}, UL_{i,2}, \cdots, UL_{i,n}$ 을 모두 결정 가능.
- i를 증가하면서 모든 $UL_{*,*}$를 $\mathcal{O}(4n(n+k))$에 계산 가능.

풀이:

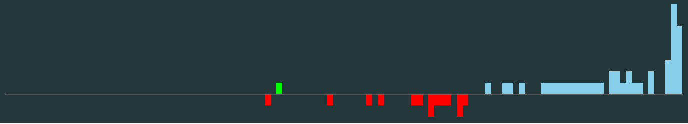
- 네 사분면 각각에 대해 집합 $UL_{*,*}$, $UR_{*,*}$,, $LL_{*,*}$, $LR_{*,*}$ 을 모두 계산.
- 각 빈 격자점에 대해, 네 집합을 사용해서 다채로운 사분면을 갖는지 판정.

복잡도: $\mathcal{O}(4^4 n^2 + nk)$.

통계량: 총 제출 201개, 정답 12개, 미공개 111개.

E: Mausoleum

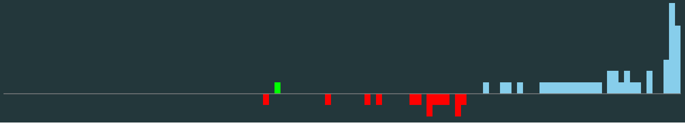
Proposer: **김수환**, Setter: **김수환**



문제: n각형의 히스토그램 내부의 점 T와 외부의 점 S가 주어진다. S에서 히스토그램의 **꼭짓점을 통해** T에 도달하는 **최단 경로의 길이**를 구하라.

E: Mausoleum

Proposer: **김수환**, Setter: **김수환**

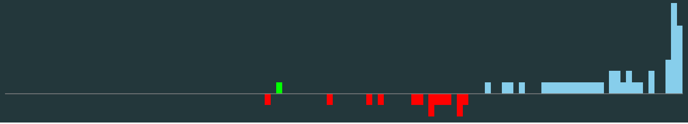


문제: n각형의 히스토그램 내부의 점 T와 외부의 점 S가 주어진다. S에서 히스토그램의 **꼭짓점을 통해** T에 도달하는 **최단 경로의 길이**를 구하라.

교양 지식: Mau · so · leum; (중요 인물 · 가문의) 묘.

E: Mausoleum

Proposer: 김수환, Setter: 김수환



문제: n각형의 히스토그램 내부의 점 T와 외부의 점 S가 주어진다. S에서 히스토그램의 **꼭짓점을 통해** T에 도달하는 **최단 경로의 길이**를 구하라.

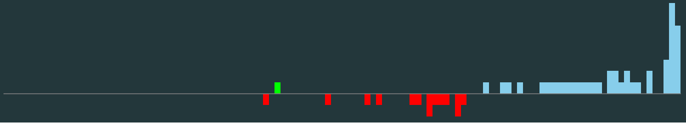
교양 지식: Mau · so · leum; (중요 인물 · 가문의) 묘.

풀이:

- T와 S에서 각 꼭짓점에 도달하는 최단 경로의 길이를 각각 $\mathrm{dist}\,(T, P_i)$, $\mathrm{dist}\,(S, P_i)$라고 하자.

E: Mausoleum

Proposer: 김수환, Setter: 김수환



문제: n각형의 히스토그램 내부의 점 T와 외부의 점 S가 주어진다. S에서 히스토그램의 **꼭짓점을 통해** T에 도달하는 **최단 경로의 길이**를 구하라.

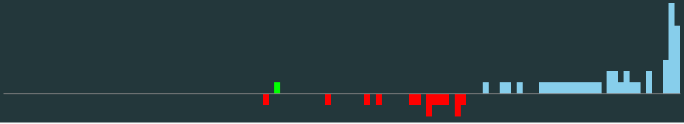
교양 지식: Mau · so · leum; (중요 인물 · 가문의) 묘.

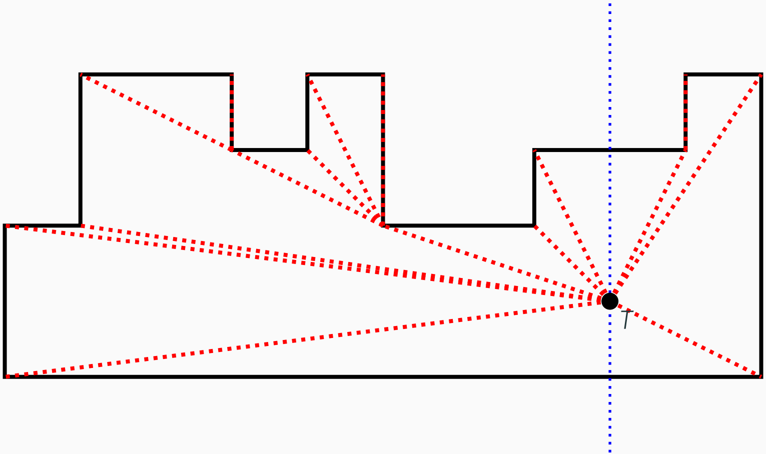
풀이:

- T와 S에서 각 꼭짓점에 도달하는 최단 경로의 길이를 각각 $\mathrm{dist}\,(T, P_i)$, $\mathrm{dist}\,(S, P_i)$라고 하자.
- 정답은 $\min_i \{\mathrm{dist}\,(T, P_i) + \mathrm{dist}\,(S, P_i)\}$

E: Mausoleum

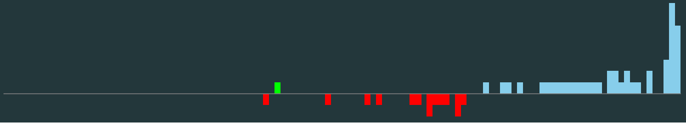
Proposer: 김수환, Setter: 김수환

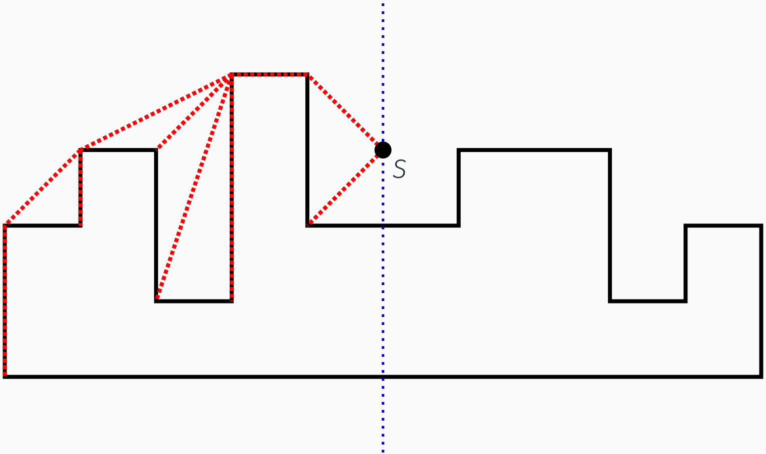




E: Mausoleum

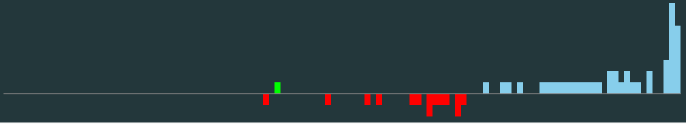
Proposer: 김수환, Setter: 김수환

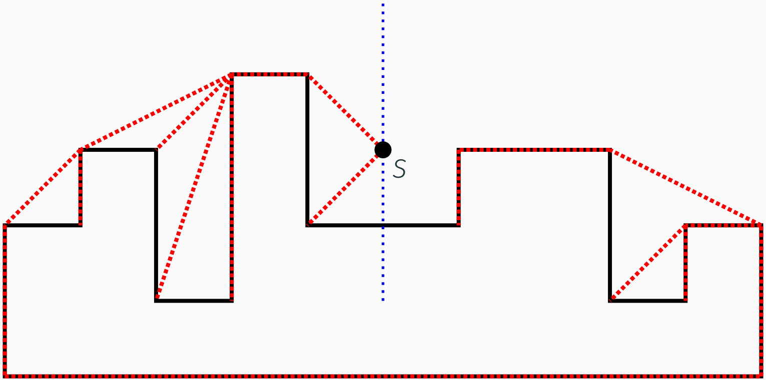




E: Mausoleum

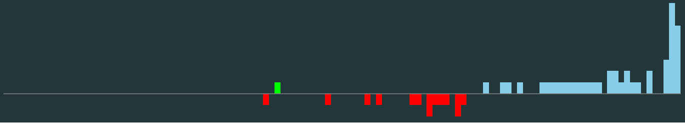
Proposer: 김수환, Setter: 김수환

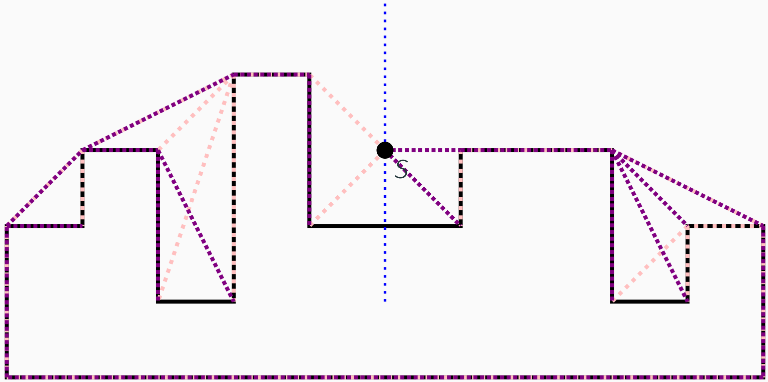




E: Mausoleum

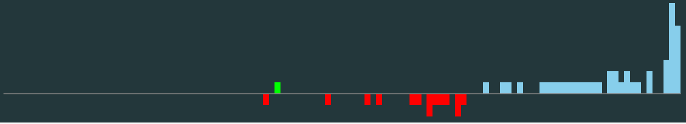
Proposer: 김수환, Setter: 김수환





E: Mausoleum

Proposer: **김수환**, Setter: **김수환**



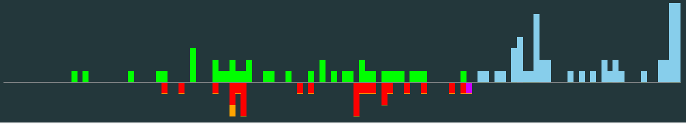
복잡도: $\mathcal{O}(n)$.

유의: 점 S가 히스토그램의 왼쪽, 오른쪽, 아래에 놓인 경우에 유의해야 합니다.

통계량: 총 제출 58개, 정답 1개, 미공개 43개.

F: Pair Sorting

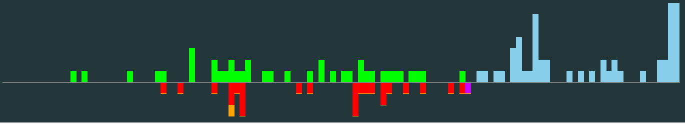
Proposer: 김재훈, Setter: 김재훈



문제: n 개의 바구니 각각에 두 개의 공이 들어있을 때, **인접한 바구니에 놓인 두 공을 바꾸는 작업**을 $0.7n^2$ 번 이하로 수행하여 모든 공의 순서를 뒤집어라.

F: Pair Sorting

Proposer: 김재훈, Setter: 김재훈



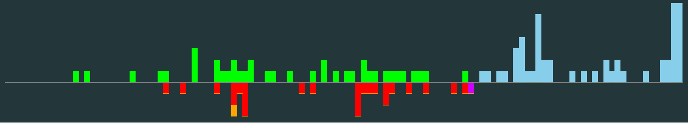
문제: n 개의 바구니 각각에 두 개의 공이 들어있을 때, **인접한 바구니에 놓인 두 공을 바꾸는 작업**을 $0.7n^2$ 번 이하로 수행하여 모든 공의 순서를 뒤집어라.

관찰?:

$$T(n) = 2\left(\frac{n}{2}\right)^2 + T\left(\frac{n}{2}\right) \implies T(n) \le \frac{2}{3}n^2$$

F: Pair Sorting

Proposer: 김재훈, Setter: 김재훈



n	$n-1$	$\cdots$	$m+2$	$m+1$	m	$\cdots$	3	2	1
n	$n-1$	$\cdots$	$m+2$	$m+1$	m	$\cdots$	3	2	1

$n-1$	$n-2$	$\cdots$	$m+1$	m	$m-1$	$\cdots$	2	1	n
n	$n-1$	$\cdots$	$m+2$	$m+1$	m	$\cdots$	3	2	1

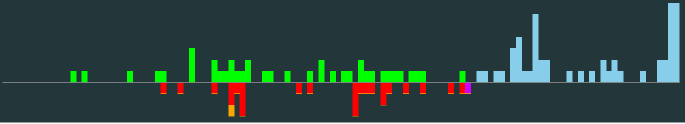
$n-1$	$n-3$	$\cdots$	m	$m-1$	$m-2$	$\cdots$	1	$n-1$	n
n	$n-2$	$\cdots$	$m+2$	$m+1$	m	$\cdots$	3	2	1

$\vdots$

$n-1$	$n-3$	$\cdots$	3	1	$m+1$	$\cdots$	$n-2$	$n-1$	n
n	$n-2$	$\cdots$	4	2	m	$\cdots$	3	2	1

F: Pair Sorting

Proposer: 김재훈, Setter: 김재훈



$\mathbf{n-1}$	$\mathbf{n-3}$	$\cdots$	$\mathbf{3}$	$\mathbf{1}$	$m+1$	$\cdots$	$n-2$	$n-1$	n
$\mathbf{n}$	$\mathbf{n-2}$	$\cdots$	$\mathbf{4}$	$\mathbf{2}$	m	$\cdots$	3	2	1

1	3	$\cdots$	$n-3$	$n-1$	$m+1$	$\cdots$	$n-2$	$n-1$	n
2	4	$\cdots$	$n-2$	n	$\mathbf{m}$	$\cdots$	$\mathbf{3}$	$\mathbf{2}$	$\mathbf{1}$

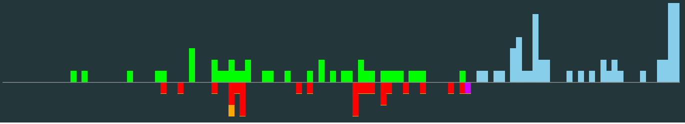
1	3	$\cdots$	$n-3$	$\mathbf{n-1}$	$m+1$	$\cdots$	$n-2$	$n-1$	n
2	4	$\cdots$	$n-2$	$\mathbf{m}$	$\mathbf{m-1}$	$\cdots$	$\mathbf{2}$	$\mathbf{1}$	n

$\vdots$

1	2	$\cdots$	$m-1$	m	$m+1$	$\cdots$	$n-2$	$n-1$	n
1	2	$\cdots$	$m-1$	m	$m+1$	$\cdots$	$n-2$	$n-1$	n

F: Pair Sorting

Proposer: 김재훈, Setter: 김재훈



풀이: n이 홀수일 때에도 유사한 전략을 사용할 수 있습니다.

통계량: 총 제출 116개, 정답 39개, 미공개 51개.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

문제: 세 가로줄 위에 놓인 n 개의 점을 **모두 덮기 위해 필요한 길이 $10\,000$ 의 정사각형의 수의 최솟값**을 구하라.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

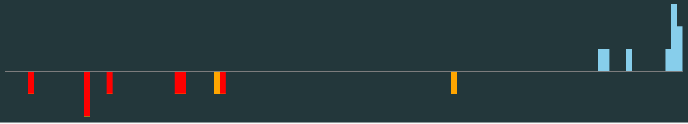
문제: 세 가로줄 위에 놓인 n 개의 점을 **모두 덮기 위해 필요한 길이** $10\,000$ **의 정사각형의 수의 최솟값**을 구하라.

풀이:

- 위, 중간, 아래에서 왼쪽 i, j, k 개의 점을 덮기 위해 필요한 정사각형의 수 $=: D_{i,j,k}$.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**



문제: 세 가로줄 위에 놓인 n 개의 점을 **모두 덮기 위해 필요한 길이** $10\,000$ **의 정사각형의 수의 최솟값**을 구하라.

풀이:

- 위, 중간, 아래에서 왼쪽 i, j, k 개의 점을 덮기 위해 필요한 정사각형의 수 $=: D_{i,j,k}$.
- 왼쪽에서 i, j, k 번째 점 중 가장 오른쪽에 위치한 점이 위 혹은 아래에 놓여있다면, 그 점을 포함하는 정사각형을 덮는 Greedy 전략이 성립.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

문제: 세 가로줄 위에 놓인 n 개의 점을 **모두 덮기 위해 필요한 길이** $10\,000$ **의 정사각형의 수의 최솟값**을 구하라.

풀이:

- 위, 중간, 아래에서 왼쪽 i, j, k 개의 점을 덮기 위해 필요한 정사각형의 수 $=: D_{i,j,k}$.
- 왼쪽에서 i, j, k 번째 점 중 가장 오른쪽에 위치한 점이 위 혹은 아래에 놓여있다면, 그 점을 포함하는 정사각형을 덮는 Greedy 전략이 성립.
- 그러한 점이 중간에 놓여있다면, 위나 아래 중 어느 방향으로 덮을 것인지 탐색해야함.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

관찰:

- 문제의 정답은 D_{n_u,n_m,n_d}.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

관찰:

- 문제의 정답은 D_{n_u,n_m,n_d}.
- 앞의 Greedy 전략대로 탐색할 때, 방문하게 되는 상태의 수는 $\mathcal{O}(n)$.
- The proof is omitted.

I: Square Stamping

Proposer: **배상원**, Setter: **배상원**

관찰:
- 문제의 정답은 D_{n_u,n_m,n_d}.
- 앞의 Greedy 전략대로 탐색할 때, 방문하게 되는 상태의 수는 $\mathcal{O}(n)$.
- The proof is omitted.

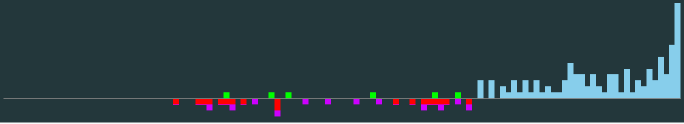
번외: n 개의 점을 x 좌표로 정렬한 후, (정사각형 최소 개수, 덮은 가장 오른쪽 점)을 관리하는 선형 DP 풀이도 존재.

복잡도: $\mathcal{O}(n \lg n)$.

통계량: 총 제출 18개, 정답 0개, 미공개 9개.

H: Protecting Kingdom

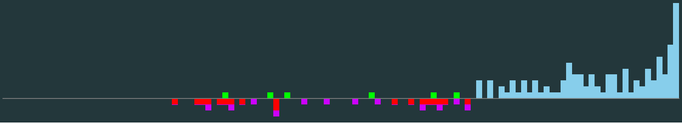
Proposer: 김재훈, Setter: 윤교준



문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다.
길이 $\le w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수를 구하라.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



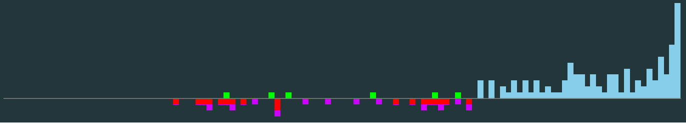
문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다. **길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수**를 구하라.

풀이:

- 정점 1을 루트로 설정.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



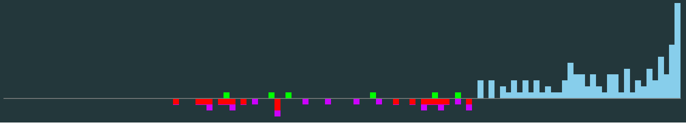
문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다.
길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수를 구하라.

풀이:

- 정점 1을 루트로 설정.
- 정점 v에서 시작해서 자식 방향으로 나아갈 때, c 개의 위험한 점을 덮는 경로의 최소 길이 $=: D_v(c)$.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



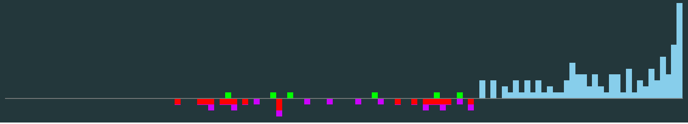
문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다.
길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수를 구하라.

풀이:

- 정점 1을 루트로 설정.
- 정점 v에서 시작해서 자식 방향으로 나아갈 때, c 개의 위험한 점을 덮는 경로의 최소 길이 $=: D_v(c)$.
- $D_{p\to v}(*)$는 $D_v(*)$에서 쉽게 계산 가능.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다. **길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수**를 구하라.

풀이:

- 정점 1을 루트로 설정.
- 정점 v에서 시작해서 자식 방향으로 나아갈 때, c 개의 위험한 점을 덮는 경로의 최소 길이 $=: D_v(c)$.
- $D_{p\to v}(*)$는 $D_v(*)$에서 쉽게 계산 가능.
- $D_{p\to v_1}(*), \cdots, D_{p\to v_g}(*)$로부터 $D_p(*)$도 쉽게 계산 가능.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준

문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다. **길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수**를 구하라.

풀이:

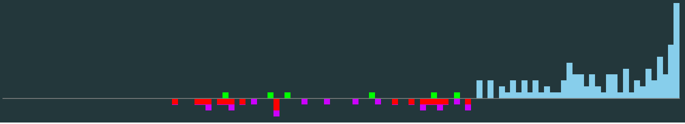
- 정점 1을 루트로 설정.
- 정점 v에서 시작해서 자식 방향으로 나아갈 때, c 개의 위험한 점을 덮는 경로의 최소 길이 $=: D_v(c)$.
- $D_{p\to v}(*)$는 $D_v(*)$에서 쉽게 계산 가능.
- $D_{p\to v_1}(*), \cdots, D_{p\to v_g}(*)$로부터 $D_p(*)$도 쉽게 계산 가능.

관찰:

- 총 연산 횟수의 총합은 $\mathcal{O}(n+k)$.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



문제: n 개의 정점으로 구성된 트리의 간선 위에 k 개의 위험한 점들이 주어진다.
길이 $\leq w$의 경로로 덮을 수 있는 위험한 점들의 최대 개수를 구하라.

풀이:

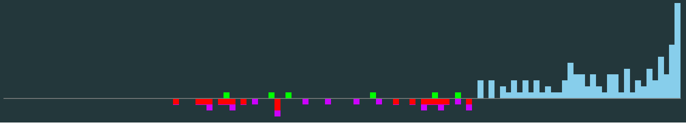
- 정점 1을 루트로 설정.
- 정점 v에서 시작해서 자식 방향으로 나아갈 때, c 개의 위험한 점을 덮는 경로의 최소 길이 $=: D_v(c)$.
- $D_{p\to v}(*)$는 $D_v(*)$에서 쉽게 계산 가능.
- $D_{p\to v_1}(*), \cdots, D_{p\to v_g}(*)$로부터 $D_p(*)$도 쉽게 계산 가능.

관찰:

- 총 연산 횟수의 총합은 $\mathcal{O}(n+k)$.
  - 각 부분트리에서 두 번째로 큰 깊이를 모두 더한 값은 $(n-1)-($트리의 깊이$)$.

H: Protecting Kingdom

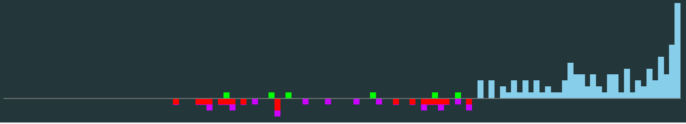
Proposer: 김재훈, Setter: 윤교준



풀이: 이제, $D_{v \to a}(*)$, $D_{v \to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



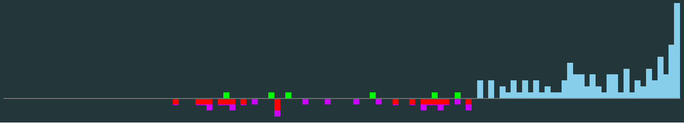
풀이: 이제, $D_{v\to a}(*)$, $D_{v\to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

관찰:

- 지금까지 구한 “덮을 수 있는 위험한 점의 수”의 최댓값을 λ라고 하자.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



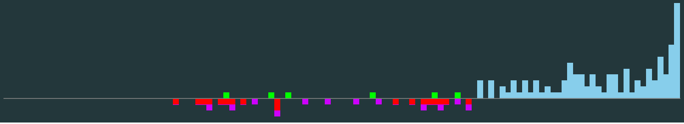
풀이: 이제, $D_{v\to a}(*)$, $D_{v\to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

관찰:

- 지금까지 구한 “덮을 수 있는 위험한 점의 수”의 최댓값을 λ라고 하자.
- $i = 1, j = \lambda$로 두고, $D_{v\to a}(i) + D_{v\to b}(j) \leq w$ 여부를 관찰하면서 두 포인터 기법.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



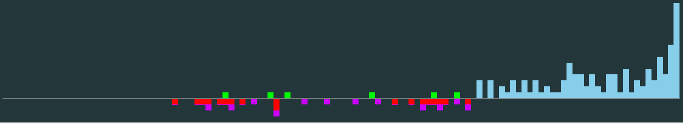
풀이: 이제, $D_{v\to a}(*)$, $D_{v\to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

관찰:

- 지금까지 구한 "덮을 수 있는 위험한 점의 수"의 최댓값을 λ라고 하자.
- $i = 1$, $j = \lambda$로 두고, $D_{v\to a}(i) + D_{v\to b}(j) \le w$ 여부를 관찰하면서 두 포인터 기법.
- 매 연산마다 i, j, λ 중 하나는 값이 진행됨.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



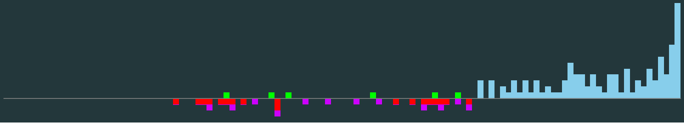
풀이: 이제, $D_{v\to a}(*)$, $D_{v\to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

관찰:

- 지금까지 구한 “덮을 수 있는 위험한 점의 수”의 최댓값을 λ라고 하자.
- $i = 1, j = \lambda$로 두고, $D_{v\to a}(i) + D_{v\to b}(j) \le w$ 여부를 관찰하면서 두 포인터 기법.
- 매 연산마다 i, j, λ 중 하나는 값이 진행됨. 고로, $\mathcal{O}\left(1 + \Delta\lambda + \min\left\{\left|D_{v\to a}\right|, \left|D_{v\to b}\right|\right\}\right)$만에 계산 가능.

H: Protecting Kingdom

Proposer: 김재훈, Setter: 윤교준



풀이: 이제, $D_{v\to a}(*)$, $D_{v\to b}(*)$가 주어질 때, 경로 $a \to v \to b$ 위에서 정점 v를 지나는 경로를 고려해야함.

관찰:
- 지금까지 구한 “덮을 수 있는 위험한 점의 수”의 최댓값을 λ라고 하자.
- $i = 1$, $j = \lambda$로 두고, $D_{v\to a}(i) + D_{v\to b}(j) \le w$ 여부를 관찰하면서 두 포인터 기법.
- 매 연산마다 i, j, λ 중 하나는 값이 진행됨. 고로, $\mathcal{O}\left(1 + \Delta\lambda + \min\left\{\left|D_{v\to a}\right|, \left|D_{v\to b}\right|\right\}\right)$만에 계산 가능.

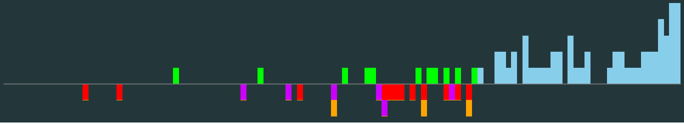
복잡도: $\mathcal{O}(n + k)$.

번외: Centroid decomposition으로 해결하기 위해서는 상당한 최적화가 요구됨.

통계량: 총 제출 151개, 정답 6개, 미공개 115개.

K: String Rank

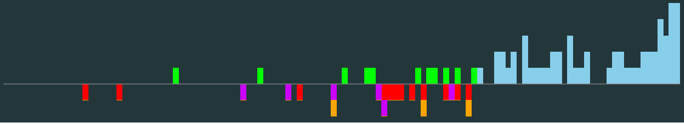
Proposer: **한요섭**, Setter: **한요섭**



문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때,
문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



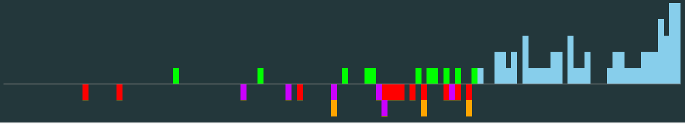
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



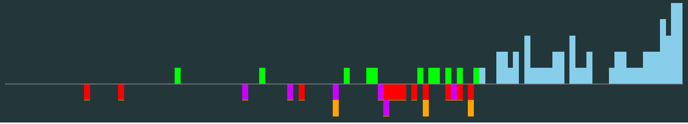
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.
- 다음과 같은 경로 $v_1 \to v_2 \to \cdots \to v_l$를 생각하자.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



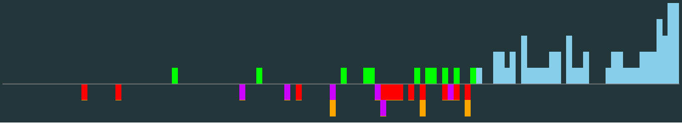
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.
- 다음과 같은 경로 $v_1 \to v_2 \to \cdots \to v_l$ 를 생각하자.
  - $n \geq v_1 > v_2 > \cdots > v_l$.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



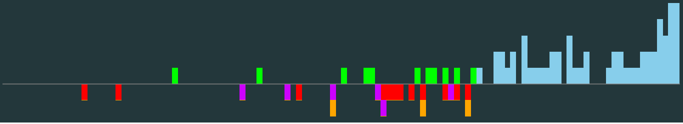
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의 $Q_k(\cdot)$가 서로 모두 다른 k의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.
- 다음과 같은 경로 $v_1 \to v_2 \to \cdots \to v_l$를 생각하자.
  - $n \geq v_1 > v_2 > \cdots > v_l$.
  - 문자열 S에서 문자 S_{v_1}가 등장하는 가장 오른쪽 위치는 v_1.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



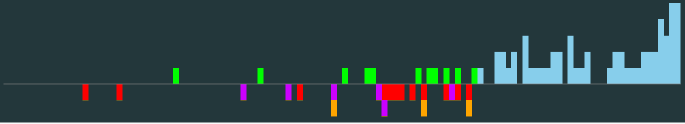
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.
- 다음과 같은 경로 $v_1 \to v_2 \to \cdots \to v_l$를 생각하자.
    - $n \geq v_1 > v_2 > \cdots > v_l$.
    - 문자열 S에서 문자 S_{v_1}가 등장하는 가장 오른쪽 위치는 v_1.
    - 문자열 S에서 문자 S_{v_2}가 등장하는, S_{v_1}의 왼쪽에 놓이면서 가장 오른쪽인 위치는 v_2.
    - $\vdots$

K: String Rank

Proposer: 한요섭, Setter: 한요섭



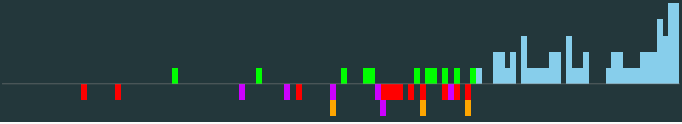
문제: 문자열의 길이 k의 부분수열을 모두 모아놓은 집합을 $Q_k(\cdot)$라고 할 때, 문자열 S의 **모든 접미사의** $Q_k(\cdot)$**가 서로 모두 다른** k**의 최솟값**을 구하라.

관찰:

- $Q_k(S[n:]) \subseteq Q_k(S[n-1:]) \subseteq \cdots \subseteq Q_k(S[2:]) \subseteq Q_k(S)$ 이므로, $Q_k(S[i:]) - Q_k(S[i+1:]) \neq \emptyset$ 여부만 판정해도 충분.
- 다음과 같은 경로 $v_1 \to v_2 \to \cdots \to v_l$를 생각하자.
  - $n \geq v_1 > v_2 > \cdots > v_l$.
  - 문자열 S에서 문자 S_{v_1}가 등장하는 가장 오른쪽 위치는 v_1.
  - 문자열 S에서 문자 S_{v_2}가 등장하는, S_{v_1}의 왼쪽에 놓이면서 가장 오른쪽인 위치는 v_2.
  - $\vdots$
- $1 \leq k \leq |S[i:]|$일 때, $Q_k(S[i:]) - Q_k(S[i+1:])$의 원소는 $v_k = i$인 경로와 대응됨.

K: String Rank

Proposer: 한요섭, Setter: 한요섭

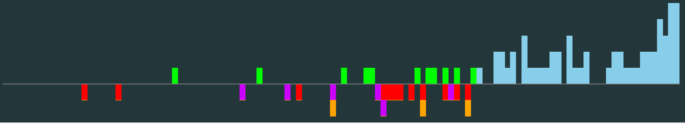


관찰:

- $v_k = i$인 경로가 존재한다면, k 이상 $|S[i:]|$ 이하의 길이를 갖는, 끝점이 i인 경로 또한 존재.

K: String Rank

Proposer: 한요섭, Setter: 한요섭

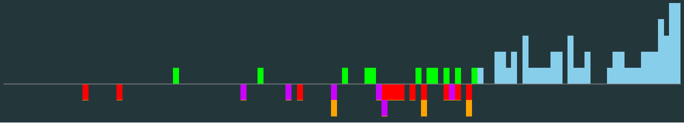


관찰:

- $v_k = i$인 경로가 존재한다면, k 이상 $|S[i:]|$ 이하의 길이를 갖는, 끝점이 i인 경로 또한 존재.
- 따라서, 각 i에 대해 끝점이 i인 최단 경로만 구하면 충분.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



관찰:

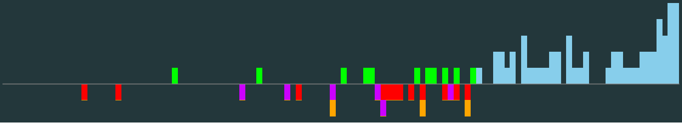
- $v_k = i$인 경로가 존재한다면, k 이상 $|S[i:]|$ 이하의 길이를 갖는, 끝점이 i인 경로 또한 존재.
- 따라서, 각 i에 대해 끝점이 i인 최단 경로만 구하면 충분.

풀이:

- 매번 가장 왼쪽으로 이동하는 Greedy 전략으로 최단 경로를 계산할 수 있음.

K: String Rank

Proposer: 한요섭, Setter: 한요섭



관찰:

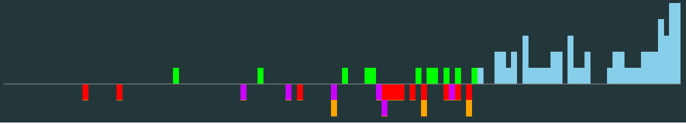
- $v_k = i$인 경로가 존재한다면, k 이상 $|S[i:]|$ 이하의 길이를 갖는, 끝점이 i인 경로 또한 존재.
- 따라서, 각 i에 대해 끝점이 i인 최단 경로만 구하면 충분.

풀이:

- 매번 가장 왼쪽으로 이동하는 Greedy 전략으로 최단 경로를 계산할 수 있음.
- 따라서, 끝점이 i인 최단 경로로부터, 끝점이 $i-1$인 최단 경로를 구성할 수 있음.

K: String Rank

Proposer: **한요섭**, Setter: **한요섭**



관찰:

- $v_k = i$인 경로가 존재한다면, k 이상 $|S[i:]|$ 이하의 길이를 갖는, 끝점이 i인 경로 또한 존재.
- 따라서, 각 i에 대해 끝점이 i인 최단 경로만 구하면 충분.

풀이:

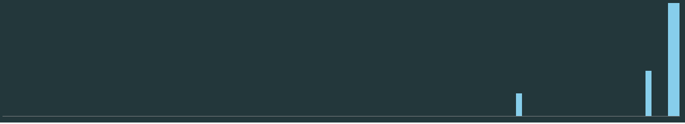
- 매번 가장 왼쪽으로 이동하는 Greedy 전략으로 최단 경로를 계산할 수 있음.
- 따라서, 끝점이 i인 최단 경로로부터, 끝점이 $i-1$인 최단 경로를 구성할 수 있음.

복잡도: $\mathcal{O}(n)$.

통계량: 총 제출 89개, 정답 11개, 미공개 57개.

D: Ladder Update

Proposer: 이승용, Setter: 이승용



문제: 사다리 타기에서 가로줄을 추가하거나 제거하는 질의가 주어진다. 각 질의에 대해, **동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수**를 구하라.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

문제: 사다리 타기에서 가로줄을 추가하거나 제거하는 질의가 주어진다. 각 질의에 대해, **동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수**를 구하라.

관찰:

- 사다리 타기는 순열 id_n 를 $\sigma \in \Sigma_n$ 으로 보내는 함수.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

문제: 사다리 타기에서 가로줄을 추가하거나 제거하는 질의가 주어진다. 각 질의에 대해, **동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수**를 구하라.

관찰:
- 사다리 타기는 순열 id_n를 $\sigma \in \Sigma_n$으로 보내는 함수.
- 동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수는 반전의 개수 $\mathrm{inv}(\sigma)$와 같음.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

문제: 사다리 타기에서 가로줄을 추가하거나 제거하는 질의가 주어진다. 각 질의에 대해, **동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수**를 구하라.

관찰:

- 사다리 타기는 순열 id_n를 $\sigma \in \Sigma_n$으로 보내는 함수.
- 동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수는 반전의 개수 $\mathrm{inv}(\sigma)$와 같음.
- 어떤 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 수가 a, b였다면, σ에서도 a와 b의 위치가 서로 바뀜.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

문제: 사다리 타기에서 가로줄을 추가하거나 제거하는 질의가 주어진다. 각 질의에 대해, **동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수**를 구하라.

관찰:

- 사다리 타기는 순열 id_n를 $\sigma \in \Sigma_n$으로 보내는 함수.
- 동치인 사다리 타기를 만들기 위해 필요한 최소 가로줄의 수는 반전의 개수 $\mathrm{inv}(\sigma)$와 같음.
- 어떤 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 수가 a, b였다면, σ에서도 a와 b의 위치가 서로 바뀜.
- σ에서 두 수의 위치를 바꾸는 질의가 주어질 때, $\mathrm{inv}(\sigma)$의 값을 구하는 작업은 $\mathcal{O}\left(q \lg^2 n + n \lg n\right)$에 계산 가능.
  - 이차원 Segment Tree
  - Offline Queries + Fenwick Tree

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.
- 사다리 타기에서 각 정수 $i \in [n]$가 이동하는 경로를 하나의 chain으로 생각.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.
- 사다리 타기에서 각 정수 $i \in [n]$가 이동하는 경로를 하나의 chain으로 생각.
- n 개의 chains은 $\mathcal{O}(n+m+q)$ 개의 정점으로 구성됨.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.
- 사다리 타기에서 각 정수 $i \in [n]$가 이동하는 경로를 하나의 chain으로 생각.
- n 개의 chains은 $\mathcal{O}(n+m+q)$ 개의 정점으로 구성됨.
- 하나의 가로줄을 추가/제거하는 작업은, 두 chains의 중간을 끊고 서로 교차하여 잇는 것과 같음.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.
- 사다리 타기에서 각 정수 $i \in [n]$가 이동하는 경로를 하나의 chain으로 생각.
- n 개의 chains은 $\mathcal{O}(n+m+q)$ 개의 정점으로 구성됨.
- 하나의 가로줄을 추가/제거하는 작업은, 두 chains의 중간을 끊고 서로 교차하여 잇는 것과 같음.

풀이:

- 각 chain을 Splay Tree로 관리.
- Chains을 끊고 잇는 작업은 Splay Tree의 기본 연산으로 처리 가능.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

관찰:

- 따라서, 가로줄이 추가/제거될 때, 그 가로줄 양끝에 오는 두 수를 효율적으로 계산하면 충분.
- 사다리 타기에서 각 정수 $i \in [n]$가 이동하는 경로를 하나의 chain으로 생각.
- n 개의 chains은 $\mathcal{O}(n + m + q)$ 개의 정점으로 구성됨.
- 하나의 가로줄을 추가/제거하는 작업은, 두 chains의 중간을 끊고 서로 교차하여 잇는 것과 같음.

풀이:

- 각 chain을 Splay Tree로 관리.
- Chains을 끊고 잇는 작업은 Splay Tree의 기본 연산으로 처리 가능.

복잡도: $\mathcal{O}\left(q \lg^2 n + n \lg n + m\right)$.

D: Ladder Update

Proposer: 이승용, Setter: 이승용

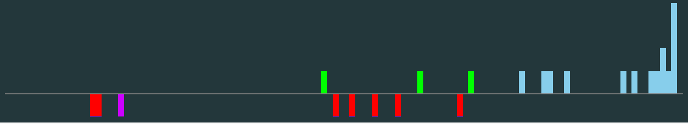
번외:

- 가로줄을 y좌표 순으로 정렬 후, 연속한 $\mathcal{O}\left(\sqrt{m+q}\right)$ 개씩 나누어 하나의 무리로 구성.
- 각 무리의 가로줄이 순열을 어떻게 바꾸는지 관리.
- $\mathcal{O}\left(q\sqrt{m+q} + q\lg^2 n + n\lg n + m\right)$
- 상당한 최적화가 요구됨.

통계량: 총 제출 13개, 정답 0개, 미공개 13개.

G: Palindromic Length

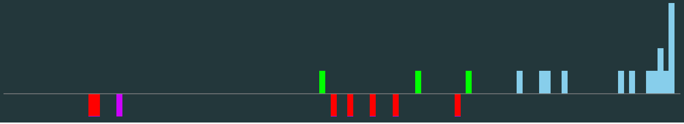
Proposer: 신찬수, Setter: 신찬수



문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



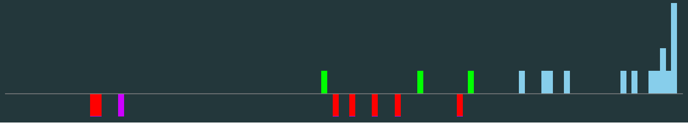
문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \leq 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



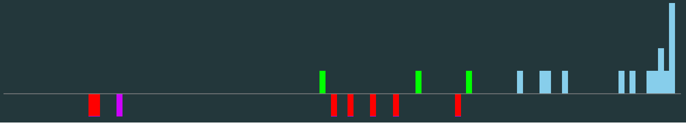
문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \le 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.
- 문자열 A에서 회문인 접미사들의 길이가 $1 = v_1 < v_2 < \cdots < v_k = |A|$라면, $v_2 - v_1 \le v_3 - v_2 \le \cdots \le v_k - v_{k-1}$는 $\mathcal{O}(\lg |A|)$ 가지의 수로 구성된 수열.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



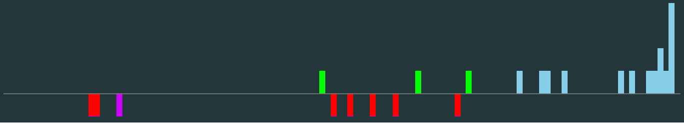
문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \leq 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.
- 문자열 A에서 회문인 접미사들의 길이가 $1 = v_1 < v_2 < \cdots < v_k = |A|$라면, $v_2 - v_1 \leq v_3 - v_2 \leq \cdots \leq v_k - v_{k-1}$는 $\mathcal{O}(\lg |A|)$ 가지의 수로 구성된 수열.
- Palindrome Tree (EerTree)를 사용하면 $v_i - v_{i-1} < v_{i+1} - v_i$인 i를 효율적으로 찾을 수 있음.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



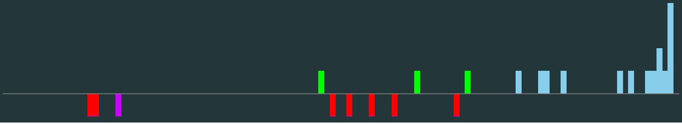
문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \leq 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.
- 문자열 A에서 회문인 접미사들의 길이가 $1 = v_1 < v_2 < \cdots < v_k = |A|$라면, $v_2 - v_1 \leq v_3 - v_2 \leq \cdots \leq v_k - v_{k-1}$는 $\mathcal{O}(\lg |A|)$ 가지의 수로 구성된 수열.
- Palindrome Tree (EerTree)를 사용하면 $v_i - v_{i-1} < v_{i+1} - v_i$인 i를 효율적으로 찾을 수 있음.
- $D_i := \mathrm{PL}(S[1 \cdots i])$라고 하면, $D_i = 1 + \min_j D_{i-v_j}$.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



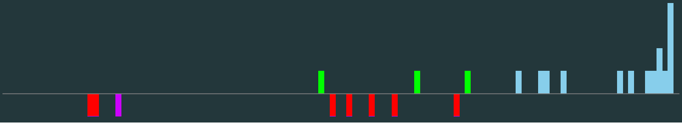
문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \le 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.
- 문자열 A에서 회문인 접미사들의 길이가 $1 = v_1 < v_2 < \cdots < v_k = |A|$라면, $v_2 - v_1 \le v_3 - v_2 \le \cdots \le v_k - v_{k-1}$는 $\mathcal{O}(\lg |A|)$ 가지의 수로 구성된 수열.
- Palindrome Tree (EerTree)를 사용하면 $v_i - v_{i-1} < v_{i+1} - v_i$인 i를 효율적으로 찾을 수 있음.
- $D_i := \mathrm{PL}(S[1 \cdots i])$라고 하면, $D_i = 1 + \min_j D_{i-v_j}$.
  - 등차수열 $v_j, v_{j+1}, \cdots, v_k$에 대해 $\min$을 효율적으로 계산 가능.

G: Palindromic Length

Proposer: 신찬수, Setter: 신찬수



문제: 문자열 S를 회문으로 분할하기 위해 필요한 **최소 분할 크기** $\mathrm{PL}(S)$을 구하라.

풀이:

- 문자열 A가 문자열 B의 접두사이자 접미사이고 $|B| \leq 2|A|$라면, A가 회문인 것은 B가 회문인 것과 동치.
- 문자열 A에서 회문인 접미사들의 길이가 $1 = v_1 < v_2 < \cdots < v_k = |A|$라면, $v_2 - v_1 \leq v_3 - v_2 \leq \cdots \leq v_k - v_{k-1}$는 $\mathcal{O}(\lg |A|)$ 가지의 수로 구성된 수열.
- Palindrome Tree (EerTree)를 사용하면 $v_i - v_{i-1} < v_{i+1} - v_i$인 i를 효율적으로 찾을 수 있음.
- $D_i := \mathrm{PL}(S[1 \cdots i])$라고 하면, $D_i = 1 + \min_j D_{i-v_j}$.
  - 등차수열 $v_j, v_{j+1}, \cdots, v_k$에 대해 $\min$을 효율적으로 계산 가능.

복잡도: $\mathcal{O}(n \lg n)$.

통계량: 총 제출 26개, 정답 3개, 미공개 15개.