2024 KAIST HAJE 프로그래밍 대회

Official Solutions

문제 리스트

#	문제	예상 난이도	출제자 / 세터
A	오버킬	Bronze	cologne
B	리듬게임	Bronze	kaistray / cologne
C	우-주 양궁	Silver	harnel_tngn / shortcakesweets
D	스네이크 게임	Gold	shortcakesweets / cologne
E	덱 조작과 쿼리	Gold	harnel_tngn
F	택틱	Platinum	harnel_tngn / kaistray
G	패러글라이딩	Platinum	harnel_tngn

A. 오버킬

- 문제에 적힌 내용을 그대로 구현하면 됩니다.
- 제한이 작아서 시간 초과 등을 고려하지 않아도 됩니다.
- 실수를 사용하면 실수 오차가 날 수 있으므로, 모든 연산을 정수로 처리하는 것이 좋습니다.

B. 리듬게임

- 변속이 없을 때, k마디를 s bpm으로 플레이하는 데 걸리는 시간은 $240k/s$입니다.
- 변속 구간을 입력 받으면서 마디의 수와 bpm을 통하여 구하면 됩니다.
- N이 10만이므로, 모든 마디에 대해서 속도를 구해주면서 더해도 시간 초과가 나지 않습니다.
- float를 사용하면 오차 범위를 만족할 수 없으므로, double을 사용하여야 합니다.

C. 우-주 양궁

- XY 평면상에서 겹치는 넓이가 있는지 판단하는 문제입니다.
  - 한 좌표에 대해, $(a, b), (c, d)$ 개구간이 겹치는지를 판단해야 합니다. $\max(a, c) < \min(b, d)$이면 두 개구간이 겹칩니다.
- X, Y에 대해 모두 겹쳤다면, 최종적으로 평면이 겹칩니다.
  - 겹친다면 $\Delta z + 1$, 아니라면 -1을 출력하면 됩니다.

D. 스네이크 게임

- 스네이크가 무한히 도는 상황 체크를 머리 위치, 방향으로 저장해 체크해야 하는 문제입니다.
- 상황 체크를 하지 않고 루프를 돌릴 경우, 이론상 판 넓이의 2배만큼 돌 수 있음에 주의해야 합니다.
- 이외 지문 상황들을 모두 정확히 구현해야 합니다. 지문을 잘못 읽어 실수하기 쉽습니다.

E. 덱 조작과 쿼리

- 덱(deck)은 사실 스택의 형태를 하고 있기 때문에, 과거 상태를 보존할 수 있으면서 push와 pop이 가능한 자료구조가 필요합니다.
- 각 행동 이후의 (덱의 가장 위에 있는 카드에 적힌 수, 덱에 있는 수의 합)을 트리 위에 올려놓는다고 생각해봅시다.

E. 덱 조작과 쿼리

- push를 할 때는 자식 노드를 만들고 pop을 할 때는 부모 노드로 올라가면 모든 과거 상태를 보존하면서 push/pop을 할 수 있습니다.
- restore k를 할 때는 k번째 행동 이후의 상태가 트리 상에서 어떤 노드를 가리키고 있는지를 찾아가면 됩니다.
- 이렇게 하면 $O(N)$ 시간에 문제를 해결할 수 있습니다.

E. 덱 조작과 쿼리

- 다른 풀이로는 오프라인 쿼리를 사용하는 방법이 있습니다.
- N개의 노드를 만든 뒤, i번째 행동에서 restore k를 한다면 k에서 i로 가는 간선을, 그렇지 않다면 $i-1$에서 i로 간선을 만들어줍니다.
- 이렇게 하면 스택을 통해서 덱에 있는 수를 관리하면서 DFS로 순회해서 모든 행동 후의 덱에 있는 수의 합을 구할 수 있습니다.
- 이 역시 $O(N)$ 시간에 문제를 해결할 수 있습니다.

E. 덱 조작과 쿼리

- 별해로 퍼시스턴트 세그먼트 트리를 사용하는 방법도 있습니다.

F. 택틱

- $N \leq 20$ 인 경우에는 브루트포스를 통해서 쉽게 해결할 수 있습니다.
- $N \leq 40$ 인 경우는 중간에서 만나기 (meet in the middle) 기법을 통해서 해결할 수 있습니다.
- 각 택틱을 절반으로 쪼개서 가능한 모든 (확률, 레이의 타임어택 점수) 의 리스트 l과 r을 구했다고 합시다.
- l에 있는 모든 원소 (p_l, s_l) 에 대해서 r에 있는 원소 (p_r, s_r) 중에서 $s_l + s_r > 0$인 원소의 p_r의 합을 계산하면, 내면의 평화를 얻을 확률을 구할 수 있습니다.
- 이는 정렬 후 투 포인터, 또는 정렬 후 이분 탐색과 누적 합을 통해서 계산할 수 있습니다.

F. 택틱

- 기댓값의 경우도 비슷하게 구할 수 있습니다.
- $\sum p_l p_r (s_l s_r)^2$ 를 구한 뒤 이를 내면의 평화를 얻을 확률로 나눠야 하는데, 여기서 p_l과 s_l을 고정하면 $p_l s_l^2 \sum p_r + 2 p_l s_l \sum p_r s_r + p_l \sum p_r s_r^2$로 전개할 수 있습니다.
- 따라서 $s_l + s_r > 0$인 원소의 $p_r, p_r s_r, p_r s_r^2$ 의 합을 관리하면서 정렬 후 투 포인터, 정렬 후 이분 탐색과 누적 합을 통해서 계산할 수 있습니다.
- 이렇게 하면 $O(N\ 2^{N/2})$ 시간에 문제를 해결할 수 있습니다.

G. 패러글라이딩

- 패러글라이더의 높이가 $f(x) = y_i - \frac{(x+x_i)^2}{c}$ 로 주어집니다.
- 실수 연산으로 뺄셈을 하는 것은 바람직하지 않으므로, $cf(x) = cy_i - (x + x_i)^2$ 를 계산한 뒤에 가장 마지막에 c로 나누면 대부분의 로직에서 정수 연산을 사용해서 실수 오차를 피할 수 있습니다.
- 우리가 관심 있는 부분은 어떤 패러글라이더가 가장 높이 날고 있느냐 이므로, 모든 패러글라이더의 $cf(x)$에 x^2를 더한 뒤에 높이를 비교해도 상관이 없습니다.

G. 패러글라이딩

- $cf(x) + x^2 = cy_i - 2xx_i - x_i$^2 는 직선이므로 컨벡스 헐 트릭 또는 리-차오 트리 등의 방법으로 특정 x에서 가장 높은 값을 갖는 직선을 찾을 수 있습니다.
- 이렇게 하면 $O(N+Q)$, 또는 $O((N+Q)\log\max(p_i))$ 시간에 문제를 해결할 수 있습니다.