2022 SCCC 봄 스터디 중급 #2

A. 15488 나이트가 체스판을 벗어나지 않을 확률

점화식을 $D(t, i, j) := t$번 이동했을 때 (i, j)에 있을 확률로 정의하면 간단한 확률 DP 문제가 됩니다.

B. 24956 나는 정말 휘파람을 못 불어

H를 기준으로 카운팅하면, (앞에 있는 W의 개수) * (뒤에서 E를 2개 이상 고르는 방법의 수)를 구하면 됩니다. $O(N)$에 해결할 수 있습니다.

C. 25047 사각형 게임 (Large)

민우가 할 수 있는 행동은 총 2^N가지입니다. 민우의 행동이 정해졌다면 종진이의 최적 행동은 $O(N^2)$에 구할 수 있으므로 $O(2^N N^2)$에 해결할 수 있습니다.

D. 2515 전시장

그림들을 높이 오름차순으로 배치해도 항상 최적해를 찾을 수 있습니다. 그림을 높이 기준으로 정렬합시다.

배치한 그림 중 가장 높은 그림이 i번째 그림일 때의 가격의 최대 합을 $D(i)$라고 정의하면, $D(i) = \max_{H_i - H_j \geq S} \{D(j) + C_i\}$입니다. 이 점화식을 그대로 구현하면 $O(N^2)$이지만, 투포인터를 이용해 i번째 그림보다 앞에 배치할 수 있는 그림을 찾고 $M(i) = \max_{1 \leq j \leq i} D_j$ 배열(Prefix Maximum)을 추가로 관리하면 $O(N)$에 계산할 수 있습니다.

E. 12900 Cheating a Boolean Tree

점화식을 $D(v, op, val) := v$를 루트로 하는 서브 트리의 연산자가 op일 때 결과를 val로 만드는데 필요한 최소 비용으로 정의하면 구현이 조금 귀찮은 트리 DP 문제가 됩니다.

F. 16885 벡터의 합

두 벡터가 서로 반대 방향 사분면에 위치하는 경우가 최적이라는 것은 쉽게 알 수 있습니다.

사분면을 마음대로 결정할 수 있으므로 모든 벡터를 1사분면으로 옮기면, 1사분면 상의 벡터 (x_1, y_1)과 3사분면으로 이동시킨 벡터 $(-x_2, -y_2)$를 더한 결과 $(x_1 - x_2, y_1 - y_2)$의 길이는 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$입니다. 결국 모든 점을 1사분면으로 옮긴 다음 가장 가까운 두 점을 구하면 문제를 해결할 수 있습니다.

G. 21100 Game

x에서 시작했을 때 얻을 수 있는 값을 $f(x)$라고 하면, $f(x) = \max\{f(x), (f(x-1) + f(x+1))/2\}$를 무한히 반복한다고 생각할 수 있고, $f(x)$는 (i, A_i)들의 Convex Hull 형태가 됩니다.

만약 (i, A_i)가 Convex Hull 위의 점이라면 정답에 $A_i \cdot N^{-1}$을 더하면 되고, 그렇지 않다면 i와 왼쪽/오른쪽에서 가장 가까운 Convex Hull 위의 점을 찾아서 확률을 계산해서 더하면 됩니다.

H. 22028 렉

Subtask 2. $M = 0$ (16점)

각 직사각형 $[x_1, x_2] \times [y_1, y_2]$에 대해 $(x_1, y_1), (x_2+1, y_2+1)$에 1을 더하고 $(x_2+1, y_1), (x_1, y_2+1)$에 -1을 더한 뒤 2차원 누적합을 구하면, (x, y)의 정답이 (x, y)에 저장됩니다. x좌표 순서대로 스위핑하면서 펜윅 트리를 이용해 y축의 누적합을 관리하면 $O((N+Q)\log X)$에 문제를 해결할 수 있습니다.

Subtask 4. 상하좌우 이동 (29점)

직사각형이 위로 이동하는 것만 생각해 봅시다. 구체적으로, $[x_1, x_2] \times [y_1, y_2]$가 y좌표가 증가하는 방향으로 d만큼 이동하는 상황을 가정하겠습니다.

Subtask 2처럼 2차원 누적합의 관점에서 생각하면, 위로 올라가는 것은 $(x_1, y_1+1 \cdots y_1+d)$와 $(x_2+1, y_2+2 \cdots y_2+d+1)$에 1을 더하고, $(x_2+1, y_1+1 \cdots y_1+d)$와 $(x_1, y_2+2 \cdots y_2+d+1)$에 -1을 더하는 것과 동일합니다. Subtask 2와 마찬가지로 x좌표 순서대로 스위핑을 하면서 y축의 누적합을 관리할 것입니다.
구간 $[s, e]$에 1을 더하고 누적합을 구하는 것은 $s, s+1, \cdots, e$에 $1, 2, \cdots, e-s+1$을 더하고, $e+1$ 이상인 지점에는 $e-s+1$을 더하는 것이라고 생각할 수 있습니다. 즉, 구간에 일차 함수를 더하는 것과 특정 값을 더하는 것으로 나눠서 생각할 수 있습니다. 펜윅 트리 2개를 이용해 할 수 있습니다.

아래로 이동하는 것은 뒤집어서 처리하면 되고, 좌우로 이동하는 것은 x, y좌표를 바꿔서 처리하면 됩니다. $O((N+M+Q)\log X)$에 해결할 수 있습니다.

Subtask 6. (100점)

$y = x$ 직선에 평행하게 이동하는 상황에서 $[1, Q_x] \times [1, Q_y]$ 영역의 합을 구하는 것은, 해당 영역에 포함되는 기울기가 1인 일차 함수들의 길이를 모두 더한 것이라고 생각할 수 있습니다.
$x - y > Q_x - Q_y \; and \, x \le Q_x$인 삼각형 영역(오른쪽 아래)과 $x - y \le Q_x - Q_y \; and \, y \le Q_y$인 삼각형 영역(왼쪽 위)으로 나눠서 생각합시다.
(x, y)를 $(y-x, x)$로 변환하면 원래 **오른쪽 아래 영역**에 있던 점이 모두 점 Q의 왼쪽 아래로 이동합니다. 마찬가지로 (x, y)를 $(x-y, y)$로 변환하면 **왼쪽 위 영역**에 있던 점이 모두 점 Q의 왼쪽 아래로 이동합니다. 각각 $y-x$와 $x-y$ 순서대로 스위핑하면 Subtask 4와 동일한 방법으로 처리할 수 있습니다.
좌표 변환 과정은 아래 텍스트에서 `I`에 주목해보세요.

```
GHI     (x-y,y)    GHI   (y-x,x)   CFI
 DEF <---------    DEF -------->   BEH
  ABC              ABC              ADG
```

$y = -x$ 직선에 평행하게 이동하는 것도 비슷하게 처리할 수 있습니다. $x + y \le Q_x + Q_y \; and \, y \le Q_y$인 사다리꼴 영역에서 $x + y \le Q_x + Q_y \; and \, x > Q_x$인 삼각형 영역을 빼면 됩니다.
(x, y)를 $(x+y, y)$로 변환하면 **사다리꼴 영역**에 있던 점이 모두 점 Q의 왼쪽 아래로 이동하고, $(x+y, -x)$로 변환하면 **삼각형 영역**에 있던 점이 모두 Q의 왼쪽 아래로 이동합니다. 각각 $x+y$ 순서대로 스위핑하면 Subtask 4와 동일한 방법으로 처리할 수 있습니다.
좌표 변환 과정은 아래 텍스트에서 `13`에 주목해보세요.

```
            1   2   3   4   5             1   2   3   4   5              21  16  11   6   1
        6   7   8   9  10     (x+y,y)    6   7   8   9  10   (x+y,-y)       22  17  12   7   2
    11  12  13  14  15      <--------  11  12  13  14  15  ---------->       23  18  13   8   
3
  16  17  18  19  20                   16  17  18  19  20                       24  19  14   
9   4
21  22  23  24  25                     21  22  23  24  25                           25  20   
15  10   5
```

전체 시간 복잡도는 $O((N + M + Q) \log N)$입니다.