GEC Cup 풀이

Official Solutions

by

GEC Cup 운영진

문제		의도한 난이도
A	특별한 학교 이름	Easy
B	특별한 작은 분수	Easy
C	특별한 학교 이름 암호화	Easy
D	특별한 큰 분수	Medium
E	특별한 드롭킥	Medium
F	특별한 서빙	Hard
G	특별한 숙제 순서 바꾸기	Hard
H	특별한 학생증	Hard
I	특별한 화재 경보	Challenging

A. 특별한 학교 이름

`implementation`

출제진 의도 – **Easy**

- 본 대회 제출: 38번, 정답 17팀 (정답률 44.74%)
- 본 대회에서 처음 푼 팀: **NLCS5** (Matthew Kang, Hyunmin Park), 2분
- 오픈 대회 제출: 291번, 정답 219명 (정답률 76.29%)
- 오픈 대회에서 처음 푼 사람: **lycoris1600**, 0분

A. 특별한 학교 이름

- `if` 문을 이용하여 입력되는 약자를 비교한 후 대응하는 정식 명칭을 출력하면 되는 문제입니다.

B. 특별한 작은 분수

`implementation`

출제진 의도 – **Easy**

- 본 대회 제출: 26번, 정답 14팀 (정답률 53.85%)
- 본 대회에서 처음 푼 팀: **NLCS5** (Matthew Kang, Hyunmin Park), 7분
- 오픈 대회 제출 179번, 정답 142명 (정답률 83.80%)
- 오픈 대회에서 처음 푼 사람: **hyperbolic**, 1분

B. 특별한 작은 분수

- N 의 최대치가 작기 때문에 $\mathcal{O}(N)$ 에 문제를 해결해도 무방합니다.
- 따라서 N 번동안 루프를 이용하여 x_1 초, x_2 초, $\cdots$, x_N 초에서의 분수의 높이를 구하면 됩니다.
- 지문에는 바닥 함수 $\left\lfloor \frac{x_t}{2} \right\rfloor$ 가 사용되어 있지만, 부동소수점 방식의 오차 때문에 실수 나눗셈과 바닥 함수를 사용해서는 안 됩니다. 대신 정수 나눗셈을 사용하여 소수점을 절사해야 합니다. 이는 D번에서도 동일합니다.

C. 특별한 학교 이름 암호화

`string, implementation`
출제진 의도 – **Easy**

- 본 대회 제출: 35번, 정답 10팀 (정답률 28.57%)
- 본 대회에서 처음 푼 팀: **I don’t know** (Daniel Kim, Jayden Lee, James Park), 36분
- 오픈 대회 제출 192번, 정답 108명 (정답률 59.90%)
- 오픈 대회에서 처음 푼 사람: **hijkl2e**, 6분

C. 특별한 학교 이름 암호화

- 경우의 수가 작기 때문에, 이름을 직접 암호화하는 프로그램을 만들어 모든 경우의 수를 하드코딩하는 방법이 유효합니다.
- 아니면, 카이사르 암호화를 하여도 두 글자의 차는 변하지 않는다는 점을 이용하여 학교의 이름을 판별하는 방법 또한 존재합니다.

D. 특별한 큰 분수

ad hoc, math
출제진 의도 – **Medium**

- 본 대회 제출: 22번, 정답 2팀 (정답률 13.64%)
- 본 대회에서 처음 푼 팀: **NLCS5** (Matthew Kang, Hyunmin Park), 138분
- 오픈 대회 제출: 154번, 정답 48명 (정답률 28.00%)
- 오픈 대회에서 처음 푼 사람: **hyperbolic**, 8분

D. 특별한 큰 분수

- 어떤 시점에서 x가 8 이상이라고 합시다. 이 경우 x는 반드시 다음과 같은 단계에 따라 바뀝니다.
  - x가 홀수인 경우: $x \to 2x \oplus 6 \to x \oplus 5 \to \left\lfloor \frac{x}{2} \right\rfloor \oplus 4$
  - x가 짝수인 경우: $x \to \frac{x}{2} \oplus 6$
- x가 8 이상일 때, x의 가장 큰 0이 아닌 비트에 해당하는 값을 M이라고 합시다. 이때 $k < 8$이라면 $\left\lfloor \frac{x}{2} \right\rfloor \oplus k < M \leq x$입니다.
- 따라서, x의 MSB(최상위 비트)는 $\mathcal{O}(1)$번 이내에 $\frac{1}{2}$로 감소합니다.
- 이러한 이유로 x는 $\mathcal{O}(1) \cdot \mathcal{O}(\log n)$ 단계 이내에 8 미만이 됩니다.

D. 특별한 큰 분수

- x 가 8 미만일 때에는 다음과 같은 3 가지 사이클 중 하나에 빠집니다.
  - $3 \to 0 \to 6 \to 5 \to 12 \to 0$
  - $1 \to 4 \to 4$
  - $2 \to 7 \to 8 \to 2$
- 따라서 값을 8 미만까지 줄인 뒤 규칙성을 활용해 8 미만에서의 값을 $\mathcal{O}(1)$ 에 구하면 총 시간복잡도 $\mathcal{O}(\log n)$ 에 문제를 풀 수 있습니다.

E. 특별한 드롭킥

`greedy`
출제진 의도 – **medium**

- 본 대회 제출: 12번, 정답 0팀 (정답률 0.00%)
- 오픈 대회 제출: 102번, 정답 15명 (정답률 20.59%)
- 오픈 대회에서 처음 푼 사람: **jthis**, 27분

E. 특별한 드롭킥

- 항상 벽 두 덩어리를 한 덩어리로 만드는 것이 가장 이득입니다.
- 따라서, 벽 덩어리들 사이의 공간이 좁은 쪽부터 벽을 세워서 두 덩어리를 이어줍니다.
- 더 이상 덩어리를 잇는 것이 불가능할 경우, 이미 존재하는 벽 옆에 벽을 추가로 짓는 것이 가장 이득입니다.
- 다만, 벽이 처음부터 하나도 없는 경우가 예외가 되어, 이 경우에는 벽 2개 이상을 세울 수 있을 경우 세우고, 벽이 이미 존재할 때의 방법을 그대로 따르면 됩니다.
- 벽 2개 이상을 세울 수 없는 경우, 벽을 하나도 세우지 않는 것이 가장 이득이 됩니다.

F. 특별한 서빙

`priority queue`
출제진 의도 – **Hard**

- 본 대회 제출: 85번, 정답 1팀 (정답률 1.18%)
- 본 대회에서 처음 푼 팀: **NLCS5** (Matthew Kang, Hyunmin Park), 29분
- 오픈 대회 제출: 81번, 정답 21팀 (정답률 25.93%)
- 오픈 대회에서 처음 푼 사람: **jthis**, 14분

F. 특별한 서빙

- 가지를 줄 수 없는 경우, 지금까지 가지를 주지 않은 학생들 중 가장 가지를 주지 않았을 때의 불만도가 큰 학생에게 가지를 주는 것이 항상 이득입니다.
- 이를 구현하기 위해, 가지를 주지 않은 학생들을 priority queue에 놓고, 가지를 줘야만 하는 상황이 나올 때마다 priority queue에서 학생을 한 명 뽑아 그 학생에게 가지를 주는 방식을 사용합니다.

G. 특별한 숙제 순서 바꾸기

ad hoc
출제진 의도 – **Hard**

- 본 대회 제출 : 4번, 정답 0팀 (정답률 0.00%)
- 오픈 대회 제출 : 19번, 정답 11명 (정답률 57.90%)
- 오픈 대회에서 처음 푼 사람 : **cozyyg**, 20분

G. 특별한 숙제 순서 바꾸기

- 우선, 숙제가 4개인 상황을 생각해 봅시다.
- 이리저리 순서를 이동시키다 보면, 숙제가 4개인 상황에서 왼쪽 또는 오른쪽 한개에는 원하는 숙제를 보낼 수 있음을 알 수 있습니다.
- 귀납법을 사용하면, 도착 수열에 연속으로 증가하는 3개의 숙제나 연속으로 감소하는 3개의 숙제가 존재할 경우, 그 숙제들을 중심으로 숙제들을 끝부터 차례차례 확정지으면 항상 원하는 원하는 수열에 도달할 수 있음을 알 수 있습니다,
- 따라서, 처음 배열과 목표 배열이 같거나, 목표 배열에 연속으로 증가/감소하는 숙제 3개가 있다면 항상 가능하고, 그 외의 경우는 연산을 한 번이라도 사용하였으면 연속으로 증가/감소하는 숙제 3개가 존재할 것이기 때문에 불가능합니다.

H. 특별한 이름표

dynamic programming
출제진 의도 – **Hard**

- 본 대회 제출: 0번, 정답 0팀
- 오픈 대회 제출: 52번, 정답 13명 (정답률 25.00%)
- 오픈 대회에서 처음 푼 사람: **jthis**, 44분

H. 특별한 이름표

- 메모이제이션 없이 DFS로 가능한 경로를 모두 세려고 시도하면 당연히 시간 초과가 납니다. 따라서 DP를 사용해야 합니다.
- 우선 포털이 없는 경우를 생각해 봅시다. $(0, 0)$에서 (i, j)까지 가는 경우의 수를 세는 배열 D_1를 생각해 봅시다.
- DP의 기저 사례 $D_1[0][0]$는 자명히 1입니다.
- 다음으로 위쪽과 왼쪽 끝의 경우를 생각해 보면, 위쪽과 왼쪽 끝의 셀로 가는 방법은 각각 오른쪽으로만 이동하거나 아래쪽으로만 이동하는 것밖에 없습니다.

H. 특별한 이름표

- 따라서 $D_1[i][0]$은 $(0,0)$에서 $(i,0$까지 벽이 없으면 1, 있으면 0입니다. $D_1[0][j]$의 경우도 비슷합니다.
- 즉 $D_1[i][0]$이나 $D_1[0][j]$는 미로 배열이 A라면 다음 점화식으로 정의할 수 있습니다.

$$D_1[i][0] = \begin{cases} 1 & \text{if } i = 0 \\ D_1[i-1][0] \land A[i][0] & \text{otherwise} \end{cases}$$

$$D_1[0][j] = \begin{cases} 1 & \text{if } j = 0 \\ D_1[0][j-1] \land A[0][j] & \text{otherwise} \end{cases}$$

H. 특별한 이름표

- 한편 다른 경우에서는 윗칸으로 이동하는 경우의 수와 아랫칸으로 이동하는 경우의 수를 더한 것입니다. 즉 $0 < i$와 $0 < j$에 대해서 다음이 성립합니다.

$$D_1[i][j] = D_1[i-1][j] + D_1[i][j-1]$$

H. 특별한 이름표

- 다음으로 포털이 있는 경우를 생각해 봅시다.
- 포털이 있는 경우의 DP배열 D_2 를 정의합시다.
- 이때 D_2 의 각 값은 D_1 과 같은 방식으로 경우의 수를 계산하되, 현재 셀에 포털이 존재하면, 이에 대한 처리를 해 줄 필요가 있습니다.
- 포털에 대한 처리는 포털의 반대쪽의 D_1 값을 더해주는 것으로 할 수 있습니다.
- 이렇게 하면 (D_1 에서) 현재 지점과 연결된 포털에 닿을 때까지 포털을 사용하지 않다가 해당 지점에서 포털을 사용했을 때의 경우의 수를 더할 수 있습니다.
- 한편 D_2 에 D_1 의 특정 값을 더하는 연산만이 유일하게 '포털을 사용하는' 동작이므로, 포털을 최대 한번 사용하는 경우의 수만이 산입됨을 보증할 수 있습니다.

- 다만 2번 예시와 같이 포털이 현재 칸의 좌측이나 상측에 위치하는 경우 포털을 사용하나 마나 칸의 목록 (순서)가 같습니다.
- 따라서 이러한 경우에는 포털 이용으로 치지 않도록 특별한 처리를 해 줄 필요가 있습니다.
- 마지막으로 $D_2[N-1][M-1]$을 출력해 주면 끝입니다.

I. 특별한 화재 경보

`segment tree`, ad hoc
출제진 의도 – **Challenging**

- 본 대회 제출: 0번, 정답 0팀
- 오픈 대회 제출: 36번, 정답 12명 (정답률 33.33%)
- 오픈 대회에서 처음 푼 사람: **xhdtlsid2**, 12분

- 문제를 읽어보면, inversion-counting이라는 잘 알려진 문제와 매우 유사함을 알 수 있습니다. 이와 관련해서는 여기에 자세히 설명되어 있습니다.
- 그러나, 동호가 할 수 있는 연산이 문제가 됩니다.
- 약간 더 생각해 보면, 인접한 두 원소를 바꾼다면, inversion은 최대 1밖에 증가하지 않는다는 것을 알 수 있습니다.
- 또한, 학생들이 완전히 내림차순으로 정렬되지 않았다면, inversion을 항상 증가시킬 수 있다는 것도 알 수 있습니다.
- 따라서, 현재 상태의 inversion을 구하고, $inversion + L$ 과 $\frac{1}{2}N(N-1)$ (내림차순으로 정렬되었을 때의 inversion값) 중 더 작은 값을 출력하면 AC를 받을 수 있습니다.