2023 Goricon 풀이

Official Solutions

문제		의도한 난이도	출제자
A	호반우가 학교에 지각한 이유 1	**Very Easy**	hamuim
B	호반우가 학교에 지각한 이유 2	**Easy**	hamuim
C	호반우가 학교에 지각한 이유 3	**Easy**	hamuim
D	호반우가 학교에 지각한 이유 4	**Medium**	hamuim
E	호반우가 학교에 지각한 이유 5	**Medium**	hamuim
F	호반우가 학교에 지각한 이유 6	**Hard**	hamuim
G	호반우가 학교에 지각한 이유 7	**Hard**	hamuim
H	호반우가 학교에 지각한 이유 8	**Hard**	hamuim

A. 호반우가 학교에 지각한 이유 1

`math`

출제진 의도 – **Very Easy**

- ✓ 출제자: `hamuim`

A. 호반우가 학교에 지각한 이유 1

- 축복을 M 번 사용했다고 생각해 봅시다.
- 호반우의 평균 스탯 수치는 $\frac{STR + DEX + INT + LUK + M}{4}$ 가 됩니다.
- 조건에 의하면 $\frac{STR + DEX + INT + LUK + M}{4} \geq N$ 을 만족해야 합니다.
- 수식을 정리하면 $\max(4N - STR - DEX - INT - LUK, 0)$ 이 답이 됩니다.
- 시간복잡도는 $\mathcal{O}(1)$ 입니다.

B. 호반우가 학교에 지각한 이유 2

`constructive`, `ad_hoc`
출제진 의도 – **Easy**

- 출제자: `hamuim`

B. 호반우가 학교에 지각한 이유 2

- ✓ 우선 두 자리수의 소수 중 일의 자리가 $0, 2, 4, 5, 6, 8$ 인 소수는 없습니다.
- ✓ 소수소수는 소수 11 을 이용하여 쉽게 만들 수 있습니다.
- ✓ $111111\cdots$ 은 모든 연속된 두 자리수가 소수입니다.

B. 호반우가 학교에 지각한 이유 2

- A가 소수임이 보장되니 A에서 시작하여 1이 마지막에 오도록 해봅시다.
- A의 일의 자리가 $1, 3, 7$이라면 뒤에 1을, 일의 자리가 9라면 뒤에 71을 붙이면 됩니다.
- B의 십의 자리가 짝수 및 5일 때는 소수소수를 만들 수 없습니다.
- 그외엔 $11, 13, 17, 19$가 소수이니 1만 계속 넣어주면 됩니다.
- 시간복잡도는 $\mathcal{O}(N)$ 입니다.

C. 호반우가 학교에 지각한 이유 3

`stack`
출제진 의도 – **Easy**

- 출제자: `hamuim`

C. 호반우가 학교에 지각한 이유 3

- ✓ 스택을 사용해 봅시다.
- ✓ 스택에는 (통나무의 길이 L, L 을 길이로 가진 통나무의 개수) 를 원소로 넣어줍니다.
- ✓ 스택의 가장 위에는 항상 제일 긴 통나무가 있도록 합니다.
- ✓ 문제를 풀기 쉽게 처음에 $(0, 0)$ 을 하나 넣어줍시다.

C. 호반우가 학교에 지각한 이유 3

- ✓ 1 번 쿼리가 들어왔을 때는 지문의 성질로 인해 항상 스택의 가장 위에 있는 통나무의 길이보다 새로 세우는 통나무의 길이가 더 깁니다.
- ✓ 따라서 1 번 쿼리에서 입력된 길이의 통나무를 스택에 하나 넣어줍니다. 개수는 1 개입니다.
- ✓ 2 번 쿼리가 들어왔을 때는 스택의 가장 위에 있는 통나무의 길이 K 를 기준으로 $\max(0, K-$ 위력$)$ 까지 스택에서 빼줍니다.
- ✓ 이때 스택에서 원소를 빼며 빠진 통나무들의 개수를 전부 더해 줍니다.
- ✓ 이후 $(\max(0, K-$ 위력$)$, 뺀 통나무들의 개수의 합$)$ 을 스택에 원소로 다시 넣어줍니다.

C. 호반우가 학교에 지각한 이유 3

- ✓ 모든 쿼리를 수행한 이후 스택에는 완성된 계단에 남아있는 통나무들과 그들의 개수가 있습니다.
- ✓ 스택에서 이들을 전부 빼내 계산하여 답을 구할 수 있습니다.
- ✓ 시간복잡도는 $\mathcal{O}(N)$ 입니다.
- ✓ 별해로 단조성을 이용해 스택을 사용하지 않는 풀이도 있습니다.

D. 호반우가 학교에 지각한 이유 4

`disjoint_set`, `greedy`
출제진 의도 – **Medium**

- 출제자: `hamuim`

D. 호반우가 학교에 지각한 이유 4

- ✓ 슬라임 그룹에 k 마리의 킹 슬라임이 있을 때 아이템을 사용한 횟수는 $k-1$ 번입니다.
- ✓ 슬라임 그룹이 k 마리의 킹 슬라임으로 합쳐지기 바로 직전을 생각해 봅시다.
- ✓ $a+b=k$ 인 a 마리, b 마리의 킹 슬라임이 있는 그룹이 있을 것이며 이들은 $(a-1)+(b-1)$ 마리의 미니 슬라임을 만들며 합쳐진 슬라임 그룹입니다.

D. 호반우가 학교에 지각한 이유 4

- 이때 a 혹은 b가 1이라면 아이템을 사용한 횟수에 포함되지 않습니다.
- 다시 말해 이들은 항상 $(a-1)+(b-1)$마리의 미니 슬라임을 만들며 합쳐지지만 킹슬라임이 1마리인 그룹은 아이템을 사용한 $k-1$번의 횟수에 포함되지 않기에 더 많은 미니 슬라임을 만들 수 있습니다.
- 그러므로 k마리의 킹 슬라임이 있는 그룹은 순서대로 $1, 2, 3, \cdots, k-1$마리의 미니 슬라임을 만들 때가 최대가 되며, 이 경우 $\frac{k(k-1)}{2}$ 마리의 미니 슬라임을 만들 수 있습니다.

D. 호반우가 학교에 지각한 이유 4

- 이제 슬롯 머신을 돌려 나온 아이템이 입력 될 때마다 분리 집합을 이용해 두 슬라임 그룹을 합쳐봅시다.
- 이미 두 슬라임이 같은 슬라임 그룹에 존재한다면 정답에 변화는 없습니다.
- 그렇지 않다면 우선 정답에서 각 슬라임 그룹의 최대로 만들 수 있는 미니 슬라임의 개수를 뺍니다.
- 그 후에 슬라임 그룹을 합치고 합쳐진 슬라임 그룹에서 최대로 만들 수 있는 미니 슬라임의 개수를 정답에 더하면 됩니다.
- 시간복잡도는 $\mathcal{O}\left(N\alpha(N)\right)$ 입니다.

E. 호반우가 학교에 지각한 이유 5

`greedy`, `sorting`
출제진 의도 – **Medium**

- 출제자: `hamuim`

E. 호반우가 학교에 지각한 이유 5

- ✓ 우선 지상은 항상 시작점 혹은 하강한 후의 지점입니다.
- ✓ 지상을 정해놓고 경로를 생각해 봅시다.
- ✓ 상승 및 하강을 높이는 같으나 방향을 오른쪽 사선 방향으로 했다고 생각합시다.
- ✓ 지상을 기준으로 왼쪽으로의 경로와 오른쪽으로의 경로가 있을 것이며 이때의 경로는 지상보다 높거나 같아야합니다.
- ✓ 또한 지상을 기준으로 왼쪽에 있는 시작점과 오른쪽에 있는 끝점은 서로 높이가 같거나 둘 중 하나가 더 높을 것입니다.

E. 호반우가 학교에 지각한 이유 5

- ✓ 시작점과 끝점의 높이가 같다면 둘 중 하나를 다른 하나로 이어붙여 시작점 혹은 끝점을 지상으로 생각 할 수 있습니다.
- ✓ 시작점과 끝점의 높이가 다르다면 높이가 낮은쪽을 높은쪽에 이어 붙이는게 메테오의 위력이 더 큽니다.
- ✓ 결국 지상은 항상 끝점 혹은 시작점이라는 것을 알 수 있습니다.

E. 호반우가 학교에 지각한 이유 5

- 시작점을 지상으로 생각하고 i 번째로 쓴 카드에 적힌 양의 정수 쌍이 a_i, b_i 라고 하겠습니다.
- N 개의 운석들의 높이를 합하면 $(a_1) + (a_1 - b_1 + a_2) + (a_1 - b_1 + a_2 - b_2 + a_3) + \cdots$ 입니다.
- 위의 식에서 $(a_1 - b_1)$ 은 $N - 1$ 개, $(a_2 - b_2)$ 는 $N - 2$ 개, $\cdots$, $(a_{N-1} - b_{N-1})$ 은 1 개가 나오게 됩니다.
- 식을 정리하면 $\sum_{i=1}^{N-1} i(a_{N-i} - b_{N-i}) + \sum_{i=1}^{N} a_i$ 가 됩니다.

E. 호반우가 학교에 지각한 이유 5

- 구한 식에서 $(a_1 - b_1), (a_2 - b_2), \cdots, (a_{N-1} - b_{N-1})$ 에 N 개의 카드들의 $(a - b)$ 를 기준으로 내림차순으로 넣어주면 최대가 됩니다.
- 만약 내림차순으로 넣었을때 경로의 최저점이 시작점보다 작아지게 된다면 반대로 끝점을 지상으로 생각하여 계산하면 됩니다.
- 메테오의 위력이 최대가 될때 지상은 항상 시작점 혹은 끝점이기에 적어도 둘 중 하나는 경로의 최저점이 됩니다.
- 시간복잡도는 $\mathcal{O}(NlogN)$ 입니다.

F. 호반우가 학교에 지각한 이유 6

`prefix_sum`, `ad_hoc`
출제진 의도 – **Hard**

- 출제자: `hamuim`

F. 호반우가 학교에 지각한 이유 6

- 관찰을 하나 해봅시다.
- 3번 연산을 몇번을 하든 1번 연산을 하는 순간 1번 연산만을 사용했을 때와 같은 결과를 얻게 됩니다.
- 그러므로 저희는 1번 연산만을 사용한 이후 3번 연산만 사용하는 방식으로 마법열쇠의 길이를 줄여나가면 됩니다.
- 2번 4번 연산도 동일하게 적용됩니다.

F. 호반우가 학교에 지각한 이유 6

- ✓ $M \geq 4$이고 연산으로 한 쪽에서 한 번에 없앨 수 있는 수의 개수는 최대 2개이므로 오른쪽과 왼쪽을 별개로 생각할 수 있습니다.
- ✓ 길이 M으로 남을 수 있는 모든 구간에 대해 최댓값을 찾아줍시다.
- ✓ 처음의 수열에서 $M-4$개의 연속된 수들은 연산에 영향을 받지 않기에 이들을 고정시키고 오른쪽, 왼쪽의 2개의 수만 구해주면 됩니다.
- ✓ 왼쪽만 생각해 보겠습니다.
- ✓ 오른쪽은 왼쪽에서 한 것을 똑같이 시행하면 됩니다.

F. 호반우가 학교에 지각한 이유 6

- 처음의 수열 $a_1, a_2, \cdots, a_N$ 에서 구간 $[s, s+M-5]$ 을 고정했다고 합시다.
- 구간의 왼쪽에 있는 $a_1, a_2, \cdots, a_{s-1}$ 을 $t_{s-1}, t_{s-2}, \cdots, t_2, t_1$ 으로 생각하겠습니다.
- 길이 M 인 마법열쇠가 수열 $b_1, b_2, \cdots, b_M$ 을 가진다고하면 왼쪽에서는 b_1, b_2 만 구해주면 됩니다.

F. 호반우가 학교에 지각한 이유 6

- ✓ 1번 연산만 계속 시행하면 $b_1 = t_2 \oplus t_3 \oplus \cdots \oplus t_{s-1}, b_2 = t_1$ 입니다.
- ✓ 1번 연산을 계속 시행하다 3번 연산을 1번 시행하면
  $b_1 = t_2 \oplus t_3, b_2 = t_1 \oplus t_4 \oplus t_5 \cdots \oplus t_{s-1}$ 입니다.
- ✓ 1번 연산을 계속 시행하다 3번 연산을 2번 시행하면
  $b_1 = t_2 \oplus t_3 \oplus t_6 \oplus t_7 \oplus t_8 \oplus \cdots \oplus t_{s-1}, b_2 = t_1 \oplus t_4 \oplus t_5$ 입니다.
- ✓ 1번 연산을 계속 시행하다 3번 연산을 3번 시행하면
  $b_1 = t_2 \oplus t_3 \oplus t_6 \oplus t_7, b_2 = t_1 \oplus t_4 \oplus t_5 \oplus t_8 \oplus t_9 \cdots \oplus t_{s-1}$ 입니다.
- ✓ 1번 연산을 계속 시행하다 3번 연산을 4번 시행하면
  $b_1 = t_2 \oplus t_3 \oplus t_6 \oplus t_7 \oplus t_{10} \oplus t_{11} \oplus t_{12} \cdots \oplus t_{s-1}, b_2 = t_1 \oplus t_4 \oplus t_5 \oplus t_8 \oplus t_9$ 입니다.
- ✓ $\cdots$

F. 호반우가 학교에 지각한 이유 6

- ✓ 위의 시행을 통해 xor 해주는 t의 인덱스 값이 규칙을 가지는 것을 알 수 있습니다.
- ✓ 3번 연산 0번에서 $b_1 = (2, \cdots), b_2 = (1)$
- ✓ 3번 연산 1번에서 $b_1 = (2, 3), b_2 = (1, 4, \cdots)$
- ✓ 3번 연산 2번에서 $b_1 = (2, 3, 6, \cdots), b_2 = (1, 4, 5)$
- ✓ 3번 연산 3번에서 $b_1 = (2, 3, 6, 7), b_2 = (1, 4, 5, 8 \cdots)$
- ✓ 3번 연산 4번에서 $b_1 = (2, 3, 6, 7, 10, \cdots), b_2 = (1, 4, 5, 8, 9)$
- ✓ $\cdots$

F. 호반우가 학교에 지각한 이유 6

- ✓ 3번 연산을 0번 했을때부터 1번, 2번, $\cdots$ 순서대로 b_1, b_2 값을 업데이트 해줄 수 있습니다.
- ✓ $s-1$ 이하의 i에 대해 t의 $[s-1, i]$ 구간안의 모든 수들을 xor 한 값은 누적합을 이용해 $\mathcal{O}(1)$에 구할 수 있습니다.
- ✓ 가능한 b_1, b_2 중 $b_1 + b_2$가 최대인 값을 찾으면 됩니다.
- ✓ 오른쪽도 똑같이 구해주면 $b_1 + b_2 + \cdots + b_M$ 의 최댓값을 찾을 수 있습니다.
- ✓ 시간복잡도는 $\mathcal{O}\left(N^2\right)$ 입니다.
- ✓ 여담으로 $1 \cdot N$ 칸으로 나누어진 종이를 양옆으로 $1, 2$칸씩 접어 겹쳐진 수들끼리 xor 하는 것으로 생각할 수도 있습니다.

G. 호반우가 학교에 지각한 이유 7

`dynamic_programming`, `tree`
출제진 의도 – **Hard**

- ✓ 출제자: `hamuim`

G. 호반우가 학교에 지각한 이유 7

- 루트를 정해 놓고 생각해봅시다.
- 루트에서 M 번 노드까지의 경로에 있는 노드를 제외한 부분 트리들은 부분 트리 내에 있는 모든 공간을 합한 하나의 노드로 생각할 수 있습니다.
- 어떤 노드 V 에서 자식 노드로 보내주는 성수의 양은 일정하고 결국 성수들이 어디에 도착하든 V 에서 해당 자식 노드에 보내준 성수의 양은 변하지 않기 때문입니다.

G. 호반우가 학교에 지각한 이유 7

- 루트를 정해놨으니 M 번 노드부터 차근차근 생각해 봅시다.
- 우선 M 번 노드에 성수가 차기 위해서는 M 번 노드와 M 번 노드의 자식들의 공간은 전부 성수로 가득차야 하며 이들의 공간의 합이 곧 필요한 성수의 양입니다.
- M 번 노드와 그 밑의 자식들이 모두 차기위한 성수의 양을 S 라 하겠습니다.
- 즉 S 는 M 번 노드를 루트로 생각하는 부분 트리의 공간의 합입니다.

G. 호반우가 학교에 지각한 이유 7

- M 번 노드의 부모 노드 P 로 올라가 봅시다.
- P 는 자식 노드 중 번호가 작은 노드와 큰 노드에 동시에 성수를 흘려보낼 것입니다.
- P 의 자식 노드 중 M 번 노드로 통하는 자식 노드보다 번호가 큰 노드의 부분 트리의 공간을 다 합한 값을 big, 번호가 작은 노드의 부분 트리의 공간을 다 합한 값을 $small$ 이라 하겠습니다.

G. 호반우가 학교에 지각한 이유 7

- ✓ $small + S \leq big$ 이면 P 에서 사용한 성수의 양은 $2(small + S)$ 입니다.
- ✓ $big + S \leq small$ 이면 P 에서 사용한 성수의 양은 $2(big + S)$ 입니다.
- ✓ 둘 다 아니라면 P 에서 사용한 성수의 양은 $small + big + S$ 입니다.
- ✓ 이제 P 에서 사용한 성수의 양을 P 의 부모에서도 처리해주며 이를 루트 노드까지 처리하면 트리를 정화하기 위해 필요한 성수의 양을 알 수 있습니다.

G. 호반우가 학교에 지각한 이유 7

- 저희가 구해야하는 것은 모든 노드가 루트일때 필요한 성수의 양이므로 우선 M 번 노드를 루트로 고정하고 M 번 노드에서 시작하여 답을 구해봅시다.
- $dp[i]$ 를 i 번 노드가 루트일때 필요한 성수의 양이라고 정의하겠습니다.
- $sum[i]$ 를 M 번 노드가 루트일때 i 번 노드와 그 밑의 하위 노드들로 구성된 부분 트리의 모든 공간의 합이라고 정의하겠습니다.
- 현재 M 번 노드에 있을 때 $dp[M] = sum[M]$ 입니다.

G. 호반우가 학교에 지각한 이유 7

- 이제 M 번의 자식 노드인 K 번 노드로 전이해봅시다.
- 이전에 루트가 정해져있을때 했던것과 똑같이 K 번 노드를 부모로 생각하고 S 가 얼마인지 알려주면 $dp[K]$ 를 구할 수 있습니다.
- 이제 K 번 노드의 자식 노드인 T 번 노드에 T 번 노드를 부모라 생각하고 루트가 정해져있을때처럼 성수의 양을 알려줍시다.

G. 호반우가 학교에 지각한 이유 7

- ✓ 우선 K 번 노드가 루트일때 $small, big$ 의 값을 구하고 T 번 노드가 $small$ 에 해당한다면 $small$ 에서 $sum[T]$ 를 빼주고 big 에 해당한다면 big 에서 $sum[T]$ 를 빼주어 필요한 성수의 양을 구할 수 있습니다.
- ✓ 위의 방법을 모든 노드에 대해 계속 전이해주면 dp 값을 구할 수 있습니다.
- ✓ 시간복잡도는 $\mathcal{O}(N)$ 입니다.

H. 호반우가 학교에 지각한 이유 8

`dynamic_programming`, `convex_hull`, `priority_queue`
출제진 의도 – **Hard**

- ✓ 출제자: `hamuim`

H. 호반우가 학교에 지각한 이유 8

- ✓ 볼록껍질을 만들때 Graham Scan 말고 Monotone Chain 이라는 방법이 있습니다.
- ✓ 해당 방법으로 위, 아래 2 개의 볼록껍질을 관리해 줄 수 있습니다.
- ✓ x 좌표가 증가함에 따라 볼록껍질을 관리해주며 2 개의 볼록껍질에서 경계에 해당하는 점과 내부에 해당하는 점을 구별할 수 있습니다.

H. 호반우가 학교에 지각한 이유 8

- $dp[i]$를 i번 행성에 도착할 수 있는 최소시간으로 정의합시다.
- $count[i]$는 i번 행성이 볼록껍질에 속하는 개수로 정의합시다.
- 위, 아래 둘 다에 속하면 $count[i] = 2$이며 둘 중 하나에만 속한다면 $count[i] = 1$, 둘 다 아니라면 $count[i] = 0$입니다.
- 2개의 우선순위 큐를 이용해 볼록껍질 내부와 경계의 dp값을 관리할 수 있습니다.
- in을 내부에 해당하는 우선순위 큐, out을 경계에 해당하는 우선순위 큐라고 정의하겠습니다.

H. 호반우가 학교에 지각한 이유 8

- ✓ i번 행성까까지 생각했을때 i번 행성을 포함한 볼록껍질을 구하며 경계에서 빠지는 점이 있다면 빠지는 점의 dp값을 in에 넣어줍니다.
- ✓ $count$배열을 이용해 볼록껍질에서 빠질때마다 -1을 더하여 $count$값이 0이 되면 볼록껍질의 경계에서 빠지는 점임을 알 수 있습니다.

H. 호반우가 학교에 지각한 이유 8

- ✓ $dp[i] = \min(\min(out) + a_i, \min(in) + b_i)$ 로 구할 수 있습니다.
- ✓ $dp[i]$ 를 구하기 전에 $\min(out), \min(in)$ 이 볼록껍질의 경계에 존재하는 것이 맞는지와 $s \le e \le s + M$ 을 만족하는지 먼저 확인해 주어야 합니다.
- ✓ 우선순위 큐에 넣을때 행성의 번호를 같이 넣어주면 위의 제한을 쉽게 판단할 수 있습니다.
- ✓ 경계에 존재하는지는 $count$ 배열의 값이 0 인지만 확인해 주면 됩니다.

H. 호반우가 학교에 지각한 이유 8

- i 번 행성까지 생각했을 때 i 번 행성은 항상 볼록껍질 경계에 위치하기에 $dp[i]$ 를 out 에 넣어줍니다.
- 처음에 $dp[i]$ 를 out 에 넣어줄때 i 번 행성은 위, 아래 볼록껍질 둘 다에 속하므로 $count[i] = 2$ 입니다.
- N 번 행성까지 계산한 후에 $dp[N]$ 이 정답이 됩니다.
- 시간복잡도는 $\mathcal{O}\left(NlogN\right)$ 입니다.