제1회 보라매컵 본선 풀이

문제		의도한 난이도	출제자
A	장기자랑	**Easy**	99asdfg
B	영내순환버스	**Easy**	asdarwin03, 99asdfg
C	조사전달	**Medium**	99asdfg
D	이기적인 목봉 체조 (Easy)	**Medium**	99asdfg
E	운전병의 딜레마	**Hard**	99asdfg
F	체육 대회	**Hard**	99asdfg
G	MVP 투표	**Challenging**	99asdfg
H	이기적인 목봉 체조 (Hard)	**Challenging**	99asdfg

A. 장기자랑

sorting, greedy, constructive

출제진 의도 – **Easy**

- 제출 109번, 정답 69명 (정답률 64.220%)
- 처음 푼 사람: **jhnah917**, 2분
- 출제자: 99asdfg

A. 장기자랑

- 배열 a를 적절히 재정렬하여 $a_1 + \sum_{i=2}^{n} max(0, a_i - a_{i-1})$의 최댓값을 구해봅시다.
- 원소들이 오름차순인 구간 $[l, r]$에 대해, $\sum_{i=l+1}^{r} max(0, a_i - a_{i-1}) = a_r - a_l$ 입니다.
- 원소들이 내림차순인 구간 $[l, r]$에 대해, $\sum_{i=l+1}^{r} max(0, a_i - a_{i-1}) = 0$ 입니다.
- 따라서, $a_1 + \sum_{i=2}^{n} max(0, a_i - a_{i-1})$의 최댓값은 연속하는 원소들이 오름차순인 모든 구간$[l, r]$에 대해 $a_1 + \sum a_r - a_l$의 최댓값과 동일합니다.

A. 장기자랑

- 만약 원소들이 오름차순인 구간 $[l, r]$에서 $r - l \geq 4$라면 해당 구간에 속하는 원소들을 재배열하여 서로 다른 두 개의 오름차순인 구간으로 나눌 수 있습니다.
- 구간 $[l, r]$을 적절히 나누면 해당 구간에 대해 실력의 합을 $a_r - a_l + a_{r-1} - a_{l+1}$로 만들 수 있으므로, 배열 a에서 원소들이 오름차순인 구간의 길이가 4 이상인 것은 최적의 순서가 아닙니다.
- 위와 동일한 방식으로 원소들이 오름차순인 구간 $[l, r]$에서 $r - l = 3$인 구간이 두 개 이상 있으면 최적이 아님도 증명할 수 있습니다.
- 즉, 배열 a는 오름차순인 모든 부분 수열 중 최대 한 개만 길이가 3이고, 나머지는 모두 길이가 2 이하가 되도록 해야 합니다.

A. 장기자랑

- 또한, 원소들이 내림차순인 구간 $[l, r]$ 에 대해서는 실력의 합이 전혀 늘지 않으므로, $r - l \geq 3$ 인 경우 그중 하나의 값을 빼서 오름차순인 구간으로 넣어주는 것이 최적임은 자명합니다.
- 따라서, 배열 a 는 길이 2 의 오름차순 수열들을 이어 붙인 형태가 되는 것이 최적입니다.
- 이때 $a_1 + \sum a_r - a_l$ 값이 최대가 되기 위해선, a_r 은 최대한 커져야 하고 a_l 은 최대한 작아져야 합니다.
- 이는 홀수번째 자리에 가장 큰 $\lceil \frac{N}{2} \rceil$ 개의 원소를, 짝수번째 자리에 가장 작은 $\lfloor \frac{N}{2} \rfloor$ 개의 원소를 배치하면 됩니다.
- 총 시간복잡도는 $\mathcal{O}(N \log N)$ 입니다.

B. 영내순환버스

`math, prefix_sum`

출제진 의도 – **Easy**

- 제출 103번, 정답 27명 (정답률 28.155%)
- 처음 푼 사람: **jhnah917**, 8분
- 출제자: `asdarwin03`, `99asdfg`

B. 영내순환버스

- 영내순환버스는 M 명의 병사가 모두 버스 이용을 마칠 때까지 계속 같은 경로를 멈추지 않고 순환합니다.
- i 번 지점에서 출발하여 N 번 지점을 거쳐 다시 i 번 지점으로 돌아올 때 걸리는 시간은 모두 같습니다. 이 시간을 "순환 시간"이라 해봅시다.
- $T[i]$ 를 1 번 지점에서 출발하여 i 번 지점에 도착하기까지 걸리는 시간으로 정의한 후 간단한 부분합으로 값을 채워넣어봅시다. 이때 시간복잡도는 $\mathcal{O}(N)$ 입니다.
- T 를 구하는 과정에서 순환 시간도 미리 계산해줍시다. 이제부터 버스가 1 번 지점에서 출발하여 다시 1 번 지점으로 돌아온 횟수를 순환 횟수 x 라 하고, 그 순환 시간을 L 이라 합시다.
- L 은 제한 조건에 의해 항상 2 이상인 정수임에 유의합시다.

B. 영내순환버스

- 병사가 r 지점에서 하차하기 위해, p 지점에서 c 시각부터 버스를 기다리는 상황을 가정해 봅시다.
- 병사가 x 번 순환한 버스를 탑승했다고 했을 때, 승차 시각은 $xL + T[p]$ 로 나타낼 수 있습니다.
- 하차하는 시각은 2 가지 경우로 나누어 나타낼 수 있습니다.
  - $p < r$ 이면, 병사는 x 번 순환한 버스에서 하차합니다.
  - $p > r$ 이면, 병사는 $(x+1)$ 번 순환한 버스에서 하차합니다.
- 따라서 이 둘을 식으로 나타내면 다음과 같습니다.
- (하차 시각) $= \begin{cases} xL + T[r] & (p < r) \\ (x+1)L + T[r] & (p > r) \end{cases}$

B. 영내순환버스

- 병사는 c시각부터 버스를 기다리고, 버스가 오는 즉시 탑승한다는 점을 이용하여 x에 대한 부등식을 세워봅시다.
- 병사가 탑승할 버스의 순환 횟수 x는, $c \leq xL + T[p]$을 만족하는 x 중 가장 작은 정수로 나타낼 수 있습니다.
- 즉 정리하면 $\frac{c - T[p]}{L}$가 정수면 $x = \frac{c - T[p]}{L}$, 정수가 아니라면 $x = \lfloor \frac{c - T[p]}{L} \rfloor + 1$임을 알 수 있습니다.
- 그리고 위 두 식은 다음 식과 동치입니다.
- $x = \lfloor \frac{c - T[p] + L - 1}{L} \rfloor \quad (\because L \geq 2)$

B. 영내순환버스

- 위에서 정리한 x의 값을 하차 시각 식에 대입하면 다음과 같이 병사의 하차 시각을 구할 수 있습니다.
- $$(\text{하차 시각}) = \begin{cases} \lfloor \dfrac{c - T[p] + L - 1}{L} \rfloor L + T[r] & (p < r) \\ \\ (\lfloor \dfrac{c - T[p] + L - 1}{L} \rfloor + 1)L + T[r] & (p > r) \end{cases}$$
- 모든 병사의 하차 시각을 $\mathcal{O}(M)$에 구할 수 있고, 그 중 최댓값을 골라 출력하면 됩니다.
- 총 시간복잡도는 $\mathcal{O}(N + M)$입니다.
- 별해로, 주어질 수 있는 모든 입력에 대해 $0 \le x \le 5 \times 10^8$을 만족함을 이용하여 순환 횟수 x를 이분 탐색으로 찾는 풀이도 있습니다.
- 이때 시간복잡도는 $\mathcal{O}\left(N + M \log(5 \times 10^8)\right)$입니다.

C. 조사전달

greedy
출제진 의도 – **Medium**

- 제출 65번, 정답 19명 (정답률 32.308%)
- 처음 푼 사람: **jhnah917**, 14분
- 출제자: 99asdfg

C. 조사전달

- 문제를 반대로 생각해 봅시다.
- 즉, "모든 경우에 대해 차출이 가능한가?"가 아닌, "차출이 불가능한 경우가 존재하는가?"를 고려해 봅시다.
- 그러면, 차출이 불가능한 경우가 하나라도 존재한다면 NO를, 하나도 존재하지 않는다면 YES를 출력하면 됩니다.

C. 조사전달

- 차출이 불가능한 경우가 발생하기 위해선, 다음의 조건 중 하나를 만족해야 합니다.
  - 사역 i 가 가능하다고 답한 인원이 b_i 명 미만이다.
  - 사역 i, 사역 j 중 하나라도 가능하다고 답한 인원이 $b_i + b_j$ 명 미만이다.
  - $\cdots$
  - 사역 $1, \cdots,$ 사역 M 중 하나라도 가능하다고 답한 인원이 $\sum_{i=1}^{M} b_i$ 명 미만이다.
- 즉, $1 \le x \le M$ 인 x 개의 사역에 대해 x 개의 사역 중 하나라도 가능하다고 답한 인원이 해당 x 개의 사역에서 필요한 인원 수보다 적으면 됩니다.

C. 조사전달

- $x = 1$ 일때 조건을 만족하기 위해선, 어떤 b_i 에 대해 $N - b_i$ 명 이상이 한 개의 사역을 제외한 나머지 모든 사역에만 가능하다고 하면 됩니다.
- 이를 풀어쓰면, 가능하다고 답한 사역의 개수가 $M - 1$ 개 이하인 사람이 $N - b_i$ 명 이상여야 한다는 것입니다.
- 이를 x 에 대해 일반화하면, x 개의 사역에 대해 필요한 인원을 s 라 할 때, 가능하다고 답한 사역의 개수가 $M - x$ 개 이하인 사람이 $N - s$ 명 이상이어야 합니다.

C. 조사전달

- 이제 가능하다고 답한 사역의 개수가 $M-x$개 이하인 사람이 $N-s$명 이상인 경우가 있는지만 확인해 주면 됩니다.
- 이때, s가 크면 클수록 위의 경우가 나올 가능성이 증가함은 자명합니다.
- 따라서 가능하다고 답한 사역의 개수 a와 사역에 필요한 인원 b를 내림차순으로 정렬한 뒤에, x를 1부터 M까지 증가시켜 주면서 $a[\sum_{i=1}^{x} b_i] \leq M-x$인지 확인해 주면 됩니다.
- 총 시간복잡도는 $\mathcal{O}(N \log N + M \log M)$입니다.

D. 이기적인 목봉 체조 (Easy)

dp

출제진 의도 – **Medium**

- 제출 24번, 정답 10명 (정답률 41.667%)
- 처음 푼 사람 : **jhnah917**, 23분
- 출제자 : 99asdfg

D. 이기적인 목봉 체조 (Easy)

- $N \leq 1\,000$ 이므로, 구간 $[l, r]$ 의 훈련병을 하나의 그룹으로 묶었을 때의 힘의 양 $f[l][r]$ 을 $\mathcal{O}\left(N^2\right)$ 의 시간복잡도로 전처리해줄 수 있습니다.
- 이 값들을 토대로 DP 테이블을 다음과 같이 정의합시다.
- $DP[j][i]$: j 번째 병사를 끝으로 i 번째 그룹이 묶였을 때의 힘의 최댓값

D. 이기적인 목봉 체조 (Easy)

- 그러면, $k \le j$ 인 모든 k 에 대해 DP 식은 다음과 같이 계산될 수 있습니다.

$$DP[j][i] = \begin{cases} f[1][j] & (i = 1) \\ max(DP[k-1][i-1] + f[k][j]) & (i > 1) \end{cases}$$

- 위 방식으로 각 그룹에 대해 $\mathcal{O}\left(N^2\right)$ 의 시간복잡도로 DP 값을 구해준 뒤, $DP[N][M]$ 을 출력해 주면 됩니다.
- 총 시간복잡도는 $\mathcal{O}\left(N^2M\right)$ 입니다.

E. 운전병의 딜레마

dijkstra, parametric_search
출제진 의도 – **Hard**

- 제출 17번, 정답 5명 (정답률 29.412%)
- 처음 푼 사람: **wbcho0504**, 37분
- 출제자: 99asdfg

E. 운전병의 딜레마

- 우선 운전병이 도로에서 걸리는 시간을 늘려서 불편도를 줄이는 동작을 못한다고 가정해 봅시다.
- 간부님이 느끼는 불편도는 사용한 도로의 불편도의 최댓값이므로, 불편도를 x 이하인 도로만을 사용한다면 간부님이 느끼는 불편도는 반드시 x 이하입니다.
- 또한, 불편도가 $x + 1$ 이하인 도로의 개수는 불편도가 x 이하인 도로의 개수보다 반드시 더 많거나 같으므로, 1 번 구역에서 N 번 구역까지 가는데 걸리는 시간은 변수 x에 대한 단조 감소 함수입니다.
- 따라서, 사용할 도로의 불편도 x에 대해 매개 변수 탐색을 할 수 있습니다.

E. 운전병의 딜레마

- 운전병이 도로에서 걸리는 시간을 늘려서 불편도를 줄이는 동작을 한다면 어떻게 될까요?
- 불편도 x 이하의 도로만을 사용한다고 하면, 모든 도로에 대해 걸리는 시간을 $t_i + max(0, s_i - x)$ 로 바꿔줄 수 있습니다.
- 이 경우, 모든 x에 대해 모든 도로를 사용할 수 있으나, x값이 증가함에 따라 일부 도로에서 걸리는 시간이 감소할 수 있습니다.
- 따라서, 운전병이 도로에서 걸리는 시간을 늘려서 불편도를 줄이는 동작을 해도 1번 구역에서 N번 구역까지 가는데 걸리는 시간은 변수 x에 대한 단조 감소함수가 됩니다.

E. 운전병의 딜레마

- 결국, 간부님이 느낄 불편도의 값 x에 대해 매개 변수 탐색을 하여 시간 T 안에 도착할 수 있는 불편도의 최솟값을 구하면 됩니다.
- 1번 구역부터 N번 구역까지 가는 데 걸리는 시간은, 모든 도로의 시간을 $t_i + max(0, s_i - x)$로 계산해 가며 다익스트라 알고리즘을 활용하면 시간복잡도 $\mathcal{O}\left(M \log N\right)$으로 계산할 수 있습니다.
- 따라서, 총 시간복잡도는 $\mathcal{O}\left(M \log N \log 10^9\right)$ 입니다.
- 시간복잡도에 비해 시간 제한이 꽤나 빡빡한 편입니다. 다익스트라 알고리즘을 비효율적으로 짜지 않도록 주의합시다.

F. 체육 대회

bruteforcing, backtracking
출제진 의도 – **Hard**

- 제출 6번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: **-**, -분
- 출제자: 99asdfg

F. 체육 대회

- 각 팀에서 병사를 각 종목에 배치하는 경우의 수는 ${}_{15}C_5 *_{10} C_5 *_5 C_5 = \dfrac{15!}{(5!)^3} = 756\,756$ 가지입니다.
- 당연히 병사들을 배치하는 모든 경우의 수에 대해 계산하면 시간 초과가 나게 되므로, 조금씩 계산량을 최적화해봅시다.
- 우선, 각각의 병사가 어떤 종목에 배치되었는지는 중요하지 않으며, 단지 한 팀에서 가질 수 있는 각 종목에서의 실력값의 조합만 알면 된다는 점에 주목합시다.
- 각 병사의 실력값은 1 이상 15 이하이므로, 한 팀에서 각 종목에서의 최대 실력의 합은 5 이상 75 이하입니다.
- 따라서, 팀 하나에서 가질 수 있는 세 가지 종목의 실력의 합의 경우의 수는 약 $71^3 = 357\,911$ 가지입니다.

F. 체육 대회

- 이때, 각 종목을 각각 종목 1, 종목 2, 종목 3이라고 할 때, 모든 종목에 대해 실력이 낮은 경우는 아예 고려하지 않아도 됩니다.
- 가령, 각 종목의 실력값이 각각 $8, 9, 10$인 경우와 $5, 7, 10$인 경우가 존재한다면, 반드시 $5, 7, 10$의 실력을 고르는 것보다 $8, 9, 10$인 경우를 고르는 것이 승리 확률이 더 높습니다.
- 따라서, 서로 다른 (종목 1, 종목 2) 쌍에 대해 종목 3이 최대인 경우만 계산해 주어도 됩니다.
- 그러므로 하나의 팀에서 봐야 하는 경우의 수는 최대 약 $71^2 = 5041$가지입니다.

- 그러나 팀이 4개이고, 따라서 모든 경우는 5041^4가지가 되므로 아직도 계산량을 줄여야 합니다.
- 우리가 알아야 하는 점은 단순히 팀 A의 승리 횟수임에 주목합시다.
- 만약 어떤 팀 A의 실력 종목 조합에 대해 팀 A가 2회 이상 승리하지 못하게 하는 단 하나의 배치라도 있다면, 팀 A의 해당 조합으로는 우승팀이 될 수 없습니다.
- 따라서, $5041^2 = 25\,411\,681$회의 연산으로 팀 A의 실력 종목 조합 하나에 대해 팀 B가 가질 수 있는 승리 또는 무승부의 경우의 수를 구해봅시다.
- 가령 팀 A의 실력 종목 조합이 $25, 30, 35$이고 팀 B의 종목 조합이 $\{20, 30, 40\}, \{21, 22, 23\}, \{30, 25, 20\}$이라고 해봅시다.
- 팀 B가 패배했다면 0, 승리하거나 무승부였다면 1으로 이진수를 표현한다면, 팀 B의 가능한 경우의 수는 $\{011\}, \{000\}, \{100\}$이 됩니다.

F. 체육 대회

- 해당 연산에 대해 나올 수 있는 총 경우의 수는 $2^3 = 8$가지 입니다.
- 위의 연산을 팀 C, 팀 D에 대해서도 해준 뒤, 모든 약 $2^9 = 512$의 경우의 수에 대해 팀 A가 팀 B, C, D를 상대로 2승 이상을 취할 수 있는지 모두 확인해 주면 됩니다.
- 이때, bit 연산을 활용한다면 모든 2^9개의 경우의 수를 순회하지 않고도 정답을 찾을 수 있습니다.
- 최대 연산량은 $756\,756 * 4 + 71^4 * 3 + 71^2 = 79\,267\,108$회입니다.
- 그러나 이는 이론적인 최대 연산량이며, 실질적으로는 1 이상 15 이하의 실력값을 가지는 병사 15명으로 계산해야 하는 경우의 수가 71^2이 되는 경우는 존재하지 않으며, 중간 계산량을 최적화를 조금만 해줘도 계산량이 많이 줄어들어 상당히 빠른 시간 안에 정답을 계산할 수 있습니다.

G. MVP 투표

`bipartite_graph`, `dp_tree`
출제진 의도 – **Challenging**

- 제출 0번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 99asdfg

G. MVP 투표

- 각 사람을 노드로 하고 투표한 사람에서 투표받은 사람으로 단방향 연결이 되어 있는 그래프로 문제 상황을 추상화해 봅시다.
- 그렇게 만들어진 그래프는 모든 노드 N 개의 outdegree가 1 인 그래프가 되므로, 하나의 사이클을 가지는 여러 연결요소들의 집합으로 그래프가 구성됩니다.
- 거짓말쟁이를 제외한 모든 사람들은 상대방의 팀에게 투표했으므로, 해당 그래프는 거짓말쟁이 한 명을 제외하면 모두 이분 그래프여야 합니다.
- 이때, 거짓말쟁이 노드가 연결하는 노드는 거짓말쟁이와 같은 팀이 되어야 합니다.
- 해당 이분 그래프로 나눠진 두 노드의 종류를 각각 팀 A 노드와 팀 B 노드라고 합시다.

G. MVP 투표

- 모든 연결 요소들은 정점과 간선 개수가 모두 같으므로, 모든 연결 요소들은 사이클을 정확히 한 개씩만 가지고 있으며 간선의 방향은 사이클이 존재하는 방향으로 이어집니다.
- 이분 그래프가 존재하려면 사이클의 크기가 홀수이면 안되므로, 각 연결 요소의 사이클 크기에 따라 거짓말쟁이 노드의 존재 여부가 바뀝니다.
  - 만약 어떤 연결 요소의 사이클의 크기가 짝수라면, 해당 연결 요소에서는 절대로 사이클 위에 거짓말쟁이 노드가 있을 수 없습니다.
  - 만약 어떤 연결 요소의 사이클의 크기가 홀수라면, 해당 연결 요소에서는 반드시 사이클 위에 거짓말쟁이 노드가 있어야 합니다.

G. MVP 투표

- 따라서, 연결 요소가 어떻게 구성되는지에 따라 거짓말쟁이 노드가 될 수 있는 후보는 다음과 같이 나뉩니다.
  - 사이클의 크기가 홀수개인 연결 요소가 없을 때: 사이클에 해당하지 않는 모든 노드
  - 사이클의 크기가 홀수개인 연결 요소가 한 개일때: 해당 연결 요소의 사이클에 해당하는 노드
  - 사이클의 크기가 홀수개인 연결 요소가 두 개 이상일때: 모순 (-1 출력)

G. MVP 투표

- 거짓말쟁이 노드 후보들을 모두 추려냈으니, 후보들 중 실제로 거짓말쟁이 노드가 될 수 있는 노드들을 찾아봅시다.
- 사이클의 크기가 홀수인 연결 요소가 한 개인 경우는 사이클에 해당하는 모든 노드가 거짓말쟁이 노드이거나, 모든 노드가 거짓말쟁이 노드가 아니어야 합니다.
- 따라서, 사이클에 속하는 각각의 노드들을 거짓말쟁이 노드라고 가정하고 직접 이분 그래프를 구성하여 팀 인원 수의 최댓값을 세 주면 됩니다.
- 이때, 각 사이클 노드들로 연결되는 다른 모든 노드들의 팀 A 노드 개수와 팀 B 노드 개수를 저장해 두면, 각 거짓말쟁이 노드 후보들에 대해 $\mathcal{O}(1)$ 만에 팀의 최대 인원이 M 명 이상인지 판별할 수 있습니다.

G. MVP 투표

- 사이클의 크기가 모두 짝수인 경우는 사이클에 해당하지 않는 모든 노드들이 거짓말쟁이 노드가 될 수 있습니다.
- 또한, 사이클의 크기가 모두 짝수이면 이분 그래프를 만들 수 있으므로, 그래프의 노드를 모두 팀 A와 팀 B로 나눠둔 뒤 각 연결 요소에 대해 팀 A와 팀 B의 인원 수를 저장해 둡시다.
- 이때, 우리가 원하는 값은 인원 수가 많은 팀의 인원 수가 M 이상인지 판별하는 것이므로, 항상 팀 A의 인원 수가 많도록 이분 그래프를 구성합시다.

G. MVP 투표

- 어떤 노드가 거짓말쟁이 노드라면, 해당 노드가 가리키는 부분 그래프의 팀이 모두 바뀝니다.
- 즉, 거짓말쟁이 노드와 해당 노드가 가리키는 노드 간의 연결을 끊었을 때 거짓말쟁이 노드가 포함하지 않는 부분 그래프 부분의 팀이 전부 바뀌게 됩니다.
- 이때, 우리가 알고 싶은 점은 해당 연결 요소의 가능한 최대 인원 수 이므로, 거짓말쟁이 노드를 포함하는 부분 그래프의 팀이 바뀌었다고 생각해도 결과적으로 동일합니다.
- 거짓말쟁이 노드가 가리키는 부분 그래프는 반드시 사이클을 포함하고 있고, 따라서 거짓말쟁이가 포함된 그래프는 반드시 트리 구조를 띱니다.
- 이 점을 활용하여, tree DP로 빠르게 거짓말쟁이 노드가 될 수 있는지 여부를 판단해 봅시다.

- 그래프의 간선을 역방향으로 두고 생각하면, 모든 노드들은 해당 노드를 포함하는 부분 트리의 루트 노드가 됩니다.
- 따라서, 사이클에 해당하는 노드부터 역방향으로, 해당 노드를 루트 노드로 하는 부분 트리의 팀 A와 팀 B 인원 수를 tree DP로 저장해 줍시다.
- 그 뒤, 가능한 모든 거짓말쟁이 노드 후보들에 대해 해당 노드를 루트로 하는 트리의 인원을 바꿔준 뒤, 가능한 한 팀의 최대 인원이 M명 이상인지 $\mathcal{O}(1)$으로 판별해줄 수 있습니다.
- 총 시간복잡도는 $\mathcal{O}(N)$입니다.

H. 이기적인 목봉 체조 (Hard)

dp, data_structures, segtree, lazy_prop

출제진 의도 – **Challenging**

- 제출 22번, 정답 4명 (정답률 18.182%)
- 처음 푼 사람: **p_ce1052**, 151분
- 출제자: 99asdfg

H. 이기적인 목봉 체조 (Hard)

- 이기적인 목봉 체조 (Easy)의 DP 수식을 그대로 가져와 봅시다.
  - $DP[j][i]$: j 번째 병사를 끝으로 i 번째 그룹이 묶였을 때의 힘의 최댓값
  - $DP[j][i] = \begin{cases} f[i][j] & (i = 1) \\ max(DP[k-1][i-1] + f[k][j]) & (i > 1) \end{cases}$
- 우선, $i = 1$ 일 때는 Easy 버전과 동일하게 문제를 해결할 수 있습니다.
- 그러나, 이제 N 제한이 $100\,000$ 으로 늘어났으므로 구간 $[l, r]$ 의 훈련병을 하나의 그룹으로 묶었을 때의 힘의 양 $f[l][r]$ 을 $\mathcal{O}\left(N^2\right)$ 의 시간복잡도로 전처리해줄 수 없습니다.
- 이를 해결하기 위해, $k \le j$ 인 $DP[j][i]$ 에 대해 다음의 두 변수를 새로 선언해 봅시다.
  - $MAXH[k] = [k, j]$ 구간 병사들의 최대 키
  - $SUMW[k] = f[k][j]$

H. 이기적인 목봉 체조 (Hard)

- 이제 j 값을 1씩 증가시켜 주면서 $MAXH[k]$ 의 값과 $SUMW[k]$ 의 값을 수정해 봅시다.
- j 번째 병사의 키를 h_j, 힘을 s_j 라 하면 다음 식이 성립합니다.
- $SUMW[k] = \begin{cases} s_j & (h_j > MAXH[k-1]) \\ SUMW[k] + s_j & (h_j = MAXH[k-1]) \end{cases}$
- 이때, $MAXH[k]$ 의 정의에 의해 $MAXH[k-1] \geq MAXH[k]$ 가 반드시 성립합니다.
- 따라서, 값 압축을 통한 $\mathcal{O}(N \log N)$ 전처리를 해 주거나, monotone하게 관리되는 stack 등의 자료구조를 활용하면 $MAXH$ 값이 h_j 보다 작은 값을 가지는 범위 $[idx2, j)$ 와, h_j 와 같은 값을 가지는 범위 $[idx1, idx2-1]$ 을 각 j 에 대해 $\mathcal{O}(1)$ 으로 찾을 수 있습니다.

H. 이기적인 목봉 체조 (Hard)

- 따라서, 다음의 쿼리를 j가 1씩 증가할때마다 수행해 주면 됩니다.
  - $SUMW[idx2, \cdots, j] = s_j$ (set query)
  - $SUMW[idx, \cdots, idx2 - 1] += s_j$ (sum query)
  - $DP[j][i] = max(DP[k-1][i-1] + SUMW[k])$ (max query)
- 위의 쿼리는 세그먼트 트리와 lazy propagation 등을 활용하여 $\mathcal{O}(\log N)$으로 수행할 수 있습니다.
- 쿼리를 모두 수행한 뒤, $DP[N][M]$ 값을 그대로 출력하면 됩니다.
- 총 시간복잡도는 $\mathcal{O}(NM \log N)$입니다.

H. 이기적인 목봉 체조 (Hard)

- 세그먼트 트리를 사용하지 않고도 문제를 해결할 수 있습니다.
- 위에서 주어진 쿼리를 세그먼트 트리로 처리하는 대신, 위의 쿼리를 조금 더 관찰해 봅시다.
- 만약 $h_j < h_{j+1}$ 인 j 가 등장한다면, $k < j$ 이며 $h_k \le h_j$ 인 모든 k 에 대해 $SUMW[k]$ 값은 더 이상 변하지 않습니다.
- 따라서, $h_j < h_{j+1}$ 인 순간 모든 $k < j, h_k \le h_j$ 인 k 에 대해 $DP[k-1][i-1] + SUMW[k]$ 의 최댓값을 저장해 두면 더 이상 해당 k 에 대해 계산하지 않아도 됩니다.
- 또한, $h_j > h_{j+1}$ 인 경우에는 어떤 $SUMW[k]$ 값도 변하지 않으므로, 단순히 지금까지 나왔던 값들 중 최댓값을 사용하여 $DP[j][i]$ 값을 계산할 수 있습니다.
- 위 방식을 multiset 및 multimap 등의 자료구조를 활용하여 구현하면 동일한 시간복잡도인 $\mathcal{O}(NM \log N)$ 으로 문제를 해결할 수 있습니다.