제5회 보라매컵 본선 풀이

Official Solutions

알고리즘 동아리 알빠노

2025년 10월 9일

문제		의도한 난이도	출제자
A	소대 배정	**Very Easy**	onjo0127
B	군꺾문자열	**Easy**	onjo0127
C	제설 작업	**Medium**	onjo0127
D	제식 훈련 1	**Medium**	onjo0127
E	가위바위보	**Hard**	onjo0127
F	제식 훈련 2	**Hard**	onjo0127
G	신병트리대대 불침번 근무	**Hard**	onjo0127

A. 소대 배정

`simulation`, `math`
출제진 의도 – **Very Easy**

- 제출 79번, 정답 25팀 (정답률 31.65%)
- 처음 푼 팀: **Aal gorithm**, 2분
- 출제자: `onjo0127`

A. 소대 배정

- 제한이 매우 작아 첫 번째로 입소한 훈련병부터 k 번째 훈련병까지 직접 배정하면서 시뮬레이션을 하면 됩니다.

A. 소대 배정

- 식을 세워서 해결할 수도 있습니다.
- 입소 순서대로 n 명씩 묶으면 하늘이는 $m = \lfloor \frac{k-1}{n} \rfloor + 1$ 번째 묶음에 속합니다.
- m 번째 묶음은 $\lfloor \frac{m-1}{b} \rfloor + 1$ 중대에 속합니다.
- m 번째 묶음은 $(m-1) \mod b + 1$ 소대에 속합니다.

B. 군꺾문자열

string, math

출제진 의도 – **Easy**

- 제출 47번, 정답 14팀 (정답률 31.92%)
- 처음 푼 팀: **I hate PS**, 6분
- 출제자: **onjo0127**

B. 군꺾문자열

- ✓ 2로 나누거나 1을 더하는 연산만 가능하므로, 주어진 복무율의 분모가 2의 거듭제곱 형태여야만 가능합니다.

B. 군꺾문자열

- 거꾸로 생각해봅시다.
- $x = \frac{a}{2^k}$ 에서 시작해서 $x = 0$을 만드는 것이 목표입니다.
- 군꺾문자열을 뒤에서부터 읽으면서,
- **군**인 경우, $x \leftarrow x - 1 \iff a \leftarrow a - 2^k$
- **꺾**인 경우, $x \leftarrow 2x \iff k \leftarrow k - 1$

B. 군꺾문자열

- **꺾**으로 k를 1씩 줄이면서 **군**으로 a에서 2^k 만큼을 뺄지 결정해주면 됩니다.
- 예를 들어 $a = 5, b = 8$이라면,
  - 꺾: $a = 5, b = 4$
  - 군: $a = 1, b = 4$
  - 꺾: $a = 1, b = 2$
  - 꺾: $a = 1, b = 1$
  - 군: $a = 0, b = 1$
- 그러므로 답은 **군꺾꺾군꺾**입니다.

B. 군꺾문자열

- 각각의 **군**이 몇 번 **꺾**였는지가 x에 합연산으로 기여한다는 사실을 관찰하면,
- $\frac{a}{b}$를 $2^{-1}, 2^{-2}, \cdots, 2^{-k}$의 합으로 표현하는 문제로 생각할 수도 있습니다.

C. 제설 작업

`greedy`
출제진 의도 – **Medium**

- 제출 30번, 정답 8팀 (정답률 26.67%)
- 처음 푼 팀: **1등 하러 왔습니다**, 18분
- 출제자: `onjo0127`

C. 제설 작업

- 결론부터 말하자면 매번 합이 가장 작은 행/열을 제설하는 것이 최적입니다.

C. 제설 작업

- 결론부터 말하자면 매번 합이 가장 작은 행/열을 제설하는 것이 최적입니다.
- 만약 시점 t에 합이 가장 작은 행/열인 O_{min}을 선택하지 않고 O_t를 선택했다면, O_{min}을 선택하도록 순서를 바꾸는 것이 이득입니다.
- $O_1 \to \cdots \to O_t \to \cdots \to O_{min} \to \cdots \to O_{N+M}$
- $O_1 \to \cdots \to O_{min} \to O_t \to \cdots \to O_{N+M}$
- 특정 행이나 열을 제설하는 비용은 시간이 지날수록 감소하기 때문입니다.

C. 제설 작업

- 각 행/열의 합을 저장해두고, 매번 아직 고르지 않은 행/열 중 합이 가장 작은 행/열을 골라줍시다.
- 제설 작업을 할 때마다 행/열들의 합들을 업데이트해주면 됩니다.
- 시간복잡도는 $O(N^2 + M^2)$ 입니다.

D. 제식 훈련 1

`math, constructive`
출제진 의도 – **Medium**

- 제출 31번, 정답 5팀 (정답률 19.36%)
- 처음 푼 팀: **뉴비입니다**, 27분
- 출제자: `onjo0127`

D. 제식 훈련 1

- 만약 한 명의 훈련병이 정확한 목표 위치에 있다면, 그 훈련병을 기준으로 간격을 e로 조정하면 결국 모든 훈련병들이 목표 위치에 있게 됩니다.
- 이 사실에 착안하여 일단 한 명만 목표 위치로 보내는 방법을 생각해봅시다.

D. 제식 훈련 1

- 편의상 $x > 0, y > 0$인 경우를 생각합시다.
- 2열 2오 훈련병을 목표 위치인 $(x + e, y + e)$로 보내려고 해봅시다.
- 첫 번째 명령으로, $1\,1\,x + e$를 사용합니다. 2열 2오 훈련병의 위치가 $(x + e, x + e)$가 됩니다.
- 두 번째 명령으로, $2\,1\,y + e$를 사용합니다. 2열 2오 훈련병의 위치가 목표 위치인 $(x + e, y + e)$가 됩니다.
- 이제 마지막 명령으로, $2\,2\,e$를 사용합니다. 모든 훈련병이 목표 위치로 가게 됩니다.

D. 제식 훈련 1

- 편의상 $x > 0, y > 0$인 경우를 생각합시다.
- 2열 2오 훈련병을 목표 위치인 $(x + e, y + e)$로 보내려고 해봅시다.
- 첫 번째 명령으로, $1\,1\,x + e$를 사용합니다. 2열 2오 훈련병의 위치가 $(x + e, x + e)$가 됩니다.
- 두 번째 명령으로, $2\,1\,y + e$를 사용합니다. 2열 2오 훈련병의 위치가 목표 위치인 $(x + e, y + e)$가 됩니다.
- 이제 마지막 명령으로, $2\,2\,e$를 사용합니다. 모든 훈련병이 목표 위치로 가게 됩니다.
- 정리하자면 첫 번째 명령으로 x좌표를 맞춰주고, 두 번째 명령으로 y좌표를 맞춰준 뒤에 마지막 명령으로 간격을 e로 맞춰주면 됩니다.

D. 제식 훈련 1

- ✓ $x < 0, y > 0$ 일 때는 1열 2오 훈련병을,
- ✓ $x < 0, y < 0$ 일 때는 1열 1오 훈련병을,
- ✓ $x > 0, y < 0$ 일 때는 2열 1오 훈련병을 목표 위치로 보내면 됩니다.

E. 가위바위보

`ad-hoc, greedy`
출제진 의도 – **Hard**

- 제출 22번, 정답 2팀 (정답률 9.09%)
- 처음 푼 팀: **뉴비입니다**, 62분
- 출제자: `onjo0127`

E. 가위바위보

- 모든 문자를 **R**로 만드는 최소 연산 횟수부터 구해봅시다.
- 그러면 다른 문자에 대한 답도 비슷하게 구할 수 있습니다.

E. 가위바위보

- 단순히 생각해보면 모든 **P**를 **S**로 만들고, 모든 **S**를 **R**로 만들면 됩니다.
- 다만 **P**가 **R**에 둘러쌓인 형태라면 **P**를 바로 제거하는 것이 불가능합니다.
- 그러므로 $\mathbf{R} \rightarrow \mathbf{P}$ 연산을 적절히 활용해서 둘러쌓인 **R**을 제거하고, 모든 **P** 묶음이 **S**와 인접하도록 만들어줘야 합니다.

E. 가위바위보

- 이제 **S**를 기준으로 문자열을 쪼개서 생각할 수 있습니다.
- i) 양쪽 끝이 **S**: **S..S**
  - **P**와 **P** 사이 중에서 **R**이 가장 많은 부분을 제외하고 나머지를 모두 **P**로 만들어주는 것이 최적입니다.
  - **SP..PR..RP..PS** 꼴이 됩니다.
- ii) 한쪽 끝만 **S**: **S..**
  - **S**와 가장 멀리 떨어진 **P** 사이의 모든 **R**을 **P**로 바꿔줘야 합니다.
  - **SP..PR..R** 꼴이 됩니다.

E. 가위바위보

- 이 과정에서 모든 **R**이 사라졌다면 불가능합니다.
- **P**가 존재하는데 **S**가 존재하지 않는 경우도 불가능합니다.
- 모든 **P**를 **S**와 연결했으므로 이제 모든 **P**를 제거하고, 모든 **S**를 제거하면 됩니다.
- 이 알고리즘에서 사용한 연산의 수가 최적임을 증명할 수 있습니다.

F. 제식 훈련 2

dp
출제진 의도 – **Hard**

- 제출 12번, 정답 0팀 (정답률 00.00%)
- 처음 푼 팀: —
- 출제자: onjo0127

F. 제식 훈련 2

- 올바른 배치를 크게 두 가지 경우로 나눌 수 있습니다.
- i) 가로 선을 하나 그어 북쪽을 바라보는 훈련병들과 남쪽을 바라보는 훈련병들을 나눌 수 있는 경우
- ii) 세로 선을 하나 그어 서쪽을 바라보는 훈련병들과 동쪽을 바라보는 훈련병들을 나눌 수 있는 경우
- 선을 긋는 것이 불가능한 경우는 없습니다.

F. 제식 훈련 2

- i)인 경우부터 살펴보면, 격자를 크게 세 가지 부분으로 나눌 수 있습니다.
- 윗부분($1, \cdots, u$번 행)에서는 훈련병들이 동, 서, 북쪽을 바라봅니다.
- 가운데 부분($u+1, \cdots, d-1$번 행)에서는 훈련병들이 동, 서쪽을 바라봅니다.
- 아랫부분($d, \cdots, N$번 행)에서는 훈련병들이 동, 서, 남쪽을 바라봅니다.

F. 제식 훈련 2

- 가운데 부분의 최소 비용은 어렵지 않게 구할 수 있습니다.
- 각 행 i에 대해서 훈련병들의 방향을 **W..WE..E**의 형태로 만드는 최소 비용 Mid_i를 계산하면 됩니다.

F. 제식 훈련 2

- 윗부분의 비용은 다음과 같이 계산할 수 있습니다.
- 모든 훈련병이 북쪽을 바라보도록 만드는 경우의 비용을 먼저 계산한 다음, 위쪽 행부터 차례대로 양옆을 서쪽과 동쪽으로 바꿔나가면서 비용을 최대한 절약한다고 생각해봅시다.
- 그러면 다음과 같이 동적 계획법(DP) 식을 세울 수 있습니다.
- $L[i][j] =$ (i 행까지의 방향을 모두 결정했고, i 행에서 1 열 - j 열 훈련병들이 서쪽을 바라보도록 했을 때 절약하는 비용의 최댓값)
- $R[i][j] =$ (i 행까지의 방향을 모두 결정했고, i 행에서 j 열 - M 열 훈련병들이 동쪽을 바라보도록 했을 때 절약하는 비용의 최댓값)

F. 제식 훈련 2

- 점화식을 세우면 다음과 같습니다.
- $L[i][j] = \max_{j \le k} L[i-1][k] +$ (i 행 1-j 열에서 초기에 서쪽을 바라보는 훈련병의 수) $-$ (i 행 1-j 열에서 초기에 북쪽을 바라보는 훈련병의 수)
- $R[i][j] = \max_{k \le j} R[i-1][k] +$ (i 행 j-M 열에서 초기에 동쪽을 바라보는 훈련병의 수) $-$ (i 행 j-M 열에서 초기에 북쪽을 바라보는 훈련병의 수)

F. 제식 훈련 2

- 그러면 $1, \cdots, i$ 번 행을 윗부분으로 두었을 때의 비용 Top_i 는 다음과 같이 계산할 수 있습니다.
- $Top_i =$ ($1, \cdots, i$ 번 행에서 북쪽을 바라보지 않는 훈련병의 수) $- \max_{l+1<r} (L[i][l] + R[i][r])$
- 이때 이 DP들을 나이브하게 계산하면 $O(NM^2)$ 의 시간이 걸리므로 부분합 배열의 적절한 사용이 필요하고, 계산 순서를 신경써줘야 합니다.

F. 제식 훈련 2

- ✓ $i, \cdots, N$ 번 행을 아랫부분으로 두었을 때의 비용 $Bottom_i$ 도 비슷하게 계산할 수 있습니다.
- ✓ 이제 $\min_{0 \le u < d \le N+1} (Top_u + Mid_{u+1} + \cdots Mid_{d-1} + Bottom_d)$ 를 계산하면 최소 비용을 계산할 수 있습니다.

F. 제식 훈련 2

- 똑같은 방식으로 ii) 경우의 최소 비용 역시 계산할 수 있습니다.
- 격자를 90도 회전한 뒤에 i)의 로직을 그대로 사용하면 구현량을 반으로 줄일 수 있습니다.

G. 신병트리대대 불침번 근무

`tree`, `dfs`
출제진 의도 – **Hard**

- 제출 11번, 정답 0팀 (정답률 00.00%)
- —
- 출제자: `onjo0127`

G. 신병트리대대 불침번 근무

- 한쪽 끝이 1번 방인 직선 구조일 때 모든 방에서 1번 방 쪽을 먼저 순찰하는 예시를 생각해보면, 총 이동 횟수가 $O(N^2)$ 스케일입니다.
- $\rightarrow$ 단순 시뮬레이션은 시간 초과를 받게 됩니다.

G. 신병트리대대 불침번 근무

- 1번 방을 루트로 고정합시다.
- 1번 방을 제외한 모든 방에서 부모 정점이 가장 마지막 순서라고 가정합니다.
- 이제 순찰을 시작하면, 이 과정이 dfs traversal과 정확히 일치한다는 것을 관찰할 수 있습니다.
- 심지어 dfs traversal이 끝나고 1번 방으로 돌아오면, 시작할 때와 정확히 같은 상태가 되어 한 번 더 순찰을 해도 똑같은 경로로 이동하게 됩니다.

G. 신병트리대대 불침번 근무

- 이번엔 나머지 방들의 세팅은 같고, k 번 방에서만 부모 정점이 중간 순서라고 가정합시다.
- 처음 dfs traversal을 진행할 때 k 번 방을 방문하면, 부모 방보다 나중 순서인 방들은 건너뛰게 됩니다. 다만 다음 번 traversal에서는 이전 슬라이드처럼 모든 자식 방들을 방문하게 됩니다.

- 이 관찰을 통해 다음과 같은 사실을 알 수 있습니다.
- i번 방을 처음 방문할 때는 i번 방의 부모 방보다 순서가 빠른 자식 방들만 방문하며, 이후에 i번 방을 다시 방문할 때는 모든 자식 방들을 방문합니다.
- 즉, 부모 방의 순서에 따라서 자식 정점들 중 일부에 일종의 가림막이 생기고, 한 번 방문하기 전에는 가림막이 걷히지 않습니다.

- 이러한 가림막이 덮는 정점들은 dfs order 상에서 구간으로 표현됩니다.
- 즉, 어떤 방이 k개의 구간에 속한다면 $k+1$번째 dfs traversal에서 그 방을 처음 방문한다는 뜻이 됩니다.

G. 신병트리대대 불침번 근무

- 이를 통해 가장 많은 구간에 속하는 방 중에서 dfs order가 가장 큰 방이 제일 마지막에 방문하는 방이 된다는 것을 알 수 있습니다.
- 이동해야 하는 총 횟수 역시 적절한 전처리와 계산을 통해 구할 수 있습니다.
- 시간복잡도는 $O(N)$ 입니다.