

INU 코드페스티벌 2025 풀이

Official Solutions

2025년 10월 1일

문제	예상 난이도	출제자
A 디딤돌 장학금	Easy	김태원 ^{ehe123}
B 알파벳 점수 계산기	Easy	방지섭 ^{bjs0306b}
C 특수한 정수 쌍	Easy	방지섭 ^{bjs0306b}
D INU 순열	Medium	허용준 ^{sk14cj}
E MBTI 소개팅	Medium	김태원 ^{ehe123}
F 햇불이의 모험	Medium	박관우 ^{gwanwoopa}
G Prime Median	Medium	허용준 ^{sk14cj}
H 건물 측량	Medium	박관우 ^{gwanwoopa}
I 브레인롯 챔피언십	Hard	김태원 ^{ehe123}
J 수열과 쿼리 2025	Hard	이동빈 ^{leedongbin}
K 터치 앤 리턴	Hard	허용준 ^{sk14cj}

A. 디딤돌 장학금

implementation

예상 난이도 - **Bronze IV** ~ **Bronze IV**

- ✓ 제출 43번, 정답 25명 (정답률 60.465%)
- ✓ 처음 푼 사람: **rev3rs1ng**, 3분
- ✓ 출제자: 김태원^{ehe123}

A. 디딤돌 장학금

- ✓ 주어진 조건을 그대로 구현하는 문제입니다.
- ✓ 학생의 분위기, 이수 학점과 성적을 입력받아 조건을 만족하는 학생 분위에 대한 장학금을 더해나가면 됩니다.

시간복잡도 : $\mathcal{O}(N)$

B. 알파벳 점수 계산기

implementation, strings

예상 난이도 - **Bronze III** ~ **Bronze II**

- ✓ 제출 38번, 정답 26명 (정답률 71.053%)
- ✓ 처음 푼 사람: **rev3rs1ng**, 6분
- ✓ 출제자: 방지섭^{bjs0306b}

B. 알파벳 점수 계산기

- ✓ 문제의 핵심은 연속된 오름차순 부분 문자열의 길이를 추적하는 것입니다.
- ✓ 문자열을 처음부터 끝까지 한 번 순회하면서 점수를 계산할 수 있습니다.
- ✓ 풀이 과정
 - 첫 번째 문자의 점수는 1 이므로, 최종 총점과 현재 점수 모두 1로 시작합니다.
 - 두 번째 문자부터 마지막 문자까지 순회합니다.
 - 현재 문자가 바로 앞 문자보다 알파벳 순서상 뒤에 온다면, 오름차순이 이어지는 것이므로 현재 점수를 이전 점수보다 1 크게 합니다.
 - 그렇지 않다면 오름차순이 끊긴 것이므로, 현재 점수는 다시 1이 됩니다.
 - 각 문자에 대해 계산된 점수를 최종 총점에 계속 더해나갑니다.

시간복잡도: $O(N)$

C. 특수한 정수 쌍

math, number theory

예상 난이도 - **Bronze I** ~ **Silver V**

- ✓ 제출 36번, 정답 15명 (정답률 41.667%)
- ✓ 처음 푼 사람: **seren**, 15분
- ✓ 출제자: 방지섭^{bj_s0306b}

C. 특수한 정수 쌍

- ✓ 문제의 핵심은 약수의 개수가 2개인 수가 소수 (prime number)라는 점을 이용하는 것입니다.
- ✓ 주어진 세 가지 조건을 종합하여 A 와 B 의 값을 N 에 대한 식으로 나타낼 수 있습니다.
- ✓ 풀이 과정
 - ' $A \times B$ 의 약수의 개수가 2개'라는 조건은 ' $A \times B$ 가 소수'라는 의미와 같습니다.
 - 두 양의 정수 A, B 의 곱이 소수가 되려면, 둘 중 하나는 반드시 1이어야 합니다.
 - ' $A < B$ ' 조건에 따라, $A = 1$ 이 되고 B 는 어떤 소수여야 합니다.
 - 이 값을 ' $B - A = N$ ' 식에 대입하면, $B - 1 = N$, 즉 $B = N + 1$ 이 됩니다.
 - 따라서 $N + 1$ 이 소수인지 아닌지만 판별하면 됩니다. $N + 1$ 이 소수일 경우, 좋은 쌍 $(1, N + 1)$ 이 유일하게 존재합니다.

시간복잡도: $\mathcal{O}(NT)$

D. INU 순열

constructive

예상 난이도 – **Silver IV** ~ **Silver III**

- ✓ 제출 45번, 정답 17명 (정답률 40.000%)
- ✓ 처음 푼 사람: **rev3rs1ng**, 16분
- ✓ 출제자: 허용준^{sk14cj}

D. INU 순열

✓ 하단에 정리한 수식과 같이 수열을 만들면 문제 조건을 충족하는 순열을 만들 수 있습니다.

$$✓ A_i = \begin{cases} (N + 1)/2, & \text{if } i = 1 \\ A_{i-1} + (-1)^i(i - 1), & \text{if } 2 \leq i \leq N \end{cases}$$

✓ 위에 있는 수식과 다른 방법으로도 문제에서 요구하는 순열을 찾을 수 있습니다.

시간복잡도: $\mathcal{O}(N)$

E. MBTI 소개팅

backtracking

예상 난이도 – **Silver III** ~ **Silver II**

- ✓ 제출 34번, 정답 1명 (정답률 2.941%)
- ✓ 처음 푼 사람: **Cu**, 176분
- ✓ 출제자: 김태원^{ehe123}

E. MBTI 소개팅

- ✓ 위 문제를 푸는 다양한 알고리즘이 있지만, $N \leq 10$ 에 주목해야 합니다.
- ✓ $N \leq 10$ 일 때, 사용할 수 있는 알고리즘의 시간 복잡도는 최대 $O(N!)$ 이 됩니다.
- ✓ 이를 통해 완전 탐색이 가능하다는 것을 알 수 있어, 순열 함수 구현에 의해 해결이 가능합니다.

- ✓ 그리디 반례 예시
 - INFP ENTP - ISTJ ESTP
 - ENTJ INFJ ISTJ ESTJ - ISTP ESTJ INFJ INTJ

시간복잡도 : $O(N!)$

F. 햇불이의 모험

implementation, simulation

예상 난이도 – **Silver III** ~ **Gold IV**

- ✓ 제출 17번, 정답 4명 (정답률 23.529%)
- ✓ 처음 푼 사람: **seren**, 114분
- ✓ 출제자: 박관우^{gwanwoopa}

F. 핫불이의 모험

- ✓ 프레임마다 키 입력이 들어와 4개의 키를 검사할 때 아래와 같이 경우의 수를 나눌 수 있습니다.
 - 현재 검사하는 키가 현재 프레임에 눌린 키와 동일할 때
 - ▶ 이전 프레임에 검사하는 키가 눌렸으면 : Stay
 - ▶ 이전 프레임에 검사하는 키가 눌리지 않았으면 : Down
 - 현재 검사하는 키가 현재 프레임에 눌린 키와 동일하지 않을 때
 - ▶ 이전 프레임에 검사하는 키가 눌렸으면 : Up
 - ▶ 이전 프레임에 검사하는 키가 눌리지 않았으면 : 이벤트 없음

- ✓ 위 조건이 참이면서 할당된 이벤트도 동일하면 핫불이가 이동하면 됩니다.

F. 햇불이의 모험

- ✓ 햇불이가 이동할 위치가 범위를 벗어나거나 장애물이 있는 칸이면 움직이지 않아야 합니다.
- ✓ 따라서 미리 이동할 칸의 좌표를 계산한 후 조건을 만족한 경우에만 햇불이를 이동시켜 줍시다.

시간 복잡도 : $\mathcal{O}(M)$

G. Prime Median

sieve of eratosthenes, binary search, dp, prefix sum

예상 난이도 – **Silver I** ~ **Gold IV**

- ✓ 제출 67번, 정답 2명 (정답률 2.985%)
- ✓ 처음 푼 사람: **Cu**, 46분
- ✓ 출제자: 허용준^{sk14cj}

G. Prime Median

- ✓ 생각하기 가장 쉬운 풀이
- ✓ 에라토스테네스의 체를 활용하여, 소수를 오름차 순으로 저장된 리스트 p 를 생성합니다.
- ✓ 이분 탐색을 이용하여, a 보다 크거나 같은 소수 중 가장 작은 값의 p 내에서의 인덱스와 b 보다 작거나 같은 소수 중 가장 큰 값의 p 내에서의 인덱스를 구합니다.
- ✓ 이 인덱스를 활용하여 $[a, b]$ 의 소수 개수를 파악할 수 있고, 문제에서 요구하는 답을 얻을 수 있습니다.

시간복잡도 : $\mathcal{O}(N \log(N))$

G. Prime Median

- ✓ 조금 더 빠른 풀이
- ✓ 에라토스테네스의 체를 활용하여, 소수를 오름차 순으로 저장된 리스트 p 를 생성합니다.
- ✓ 동적 계획법을 사용하면, 2부터 100만까지의 정수 i 보다 크거나 같은 소수 중 가장 작은 값, 정수 i 보다 작거나 같은 소수 중 가장 큰 값의 소수의 p 내에서의 인덱스를 미리 저장해둘 수 있습니다.
- ✓ 따라서, 동적 계획법을 사용하면, 구간 하나 당 $\mathcal{O}(1)$ 로 답을 구할 수 있습니다.

시간복잡도: $\mathcal{O}(N)$

H. 건물 측량

graph traversal , prefix sum

예상 난이도 - **Silver I** ~ **Gold IV**

- ✓ 제출 4번, 정답 1명 (정답률 25.000%)
- ✓ 처음 푼 사람: **Cu**, 75분
- ✓ 출제자: 박관우^{gwanwoopa}

H. 건물 측량

- ✓ 주어진 지도로는 0이 땅인지 건물인지 빠르게 파악하기 어렵습니다. 따라서 주어진 지도를 땅과 건물을 명확히 구분할 수 있는 지도로 바꿔야 합니다.
- ✓ 0으로 표현된 칸 중 상하좌우 인접한 0으로 이동해 지도의 테두리에 도달할 수 있는 칸이 땅이고, 지도의 테두리는 항상 땅이므로 지도의 테두리에서부터 탐색을 시작해봅시다.
- ✓ 그래프 탐색을 진행하면서 방문한 칸을 모두 체크합니다. 그러면 체크한 칸이 모두 땅이고 체크하지 않은 칸은 건물이 됩니다.
- ✓ 이후의 계산을 편리하게 하기 위해 땅을 0으로, 건물을 1로 저장해줍니다.

H. 건물 측량

- ✓ 이제 지도를 통해 땅과 건물을 명확히 구분할 수 있으니 고객들에게 질문을 받아봅시다.
- ✓ 질문이 들어왔을 때 주어진 범위의 합이 0이면 새 건물을 지을 수 있고, 0이 아니면 해당 값만큼 건물이 존재해 새 건물을 지을 수 없습니다.
- ✓ 이때 단순히 이중 반복문을 통해 시작 칸부터 끝 칸까지의 합을 구하면 모든 질문을 처리할 때 시간복잡도가 $O(NMQ)$ 가 되어 시간초과가 발생합니다.
- ✓ 따라서 미리 누적합 알고리즘을 사용해 전체 구간의 합을 계산해놓아야 시간안에 모든 질문을 처리할 수 있습니다.
- ✓ 이차원 누적합을 사용하거나, 각 행에 대해 일차원 누적합을 계산한 뒤 반복문을 이용해 합을 구해도 됩니다.

시간복잡도 : $O(Q)$

I. 브레인롯 챔피언십

topology_sort

예상 난이도 - Gold IV ~ Gold II

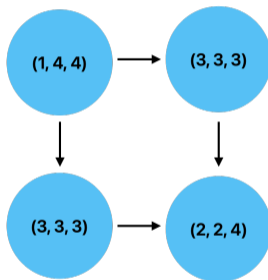
- ✓ 제출 3번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: -, -분
- ✓ 출제자: 김태원^{ehe123}

I. 브레인롯 챔피언십

- ✓ 위 문제는 두가지 문제를 해결해야 합니다.
 - 모두에게 지지 않는 캐릭터의 목록을 구하기
 - 사이클의 여부 구하기

I. 브레인롯 챔피언십

- ✓ 모두에게 지지 않는 캐릭터 목록을 알아내 봅시다.
- ✓ 각 캐릭터별 승/패를 구하여 승리한 캐릭터의 승 -> 패 방향으로 방향 그래프를 만듭니다.



- ✓ 아무에게도 화살표를 받지 않은 캐릭터가 모두에게 지지 않는 캐릭터입니다.

I. 브레인롯 챔피언십

- ✓ 이제 사이클을 구해야 합니다. 다양한 방법이 있지만, 위 풀이에서는 위상 정렬을 이용합니다.
- ✓ 화살표를 받지 않은 모든 캐릭터들을 시작으로 위상정렬을 했음에도 남는 캐릭터가 있다면 사이클이 존재한다고 판단합니다.
- ✓ 사이클이 존재하지 않으면 캐릭터 목록(궁극의 승리자 목록)을 출력하고, 존재하면 Paradoxe Absurdo를 출력하면 됩니다.

시간복잡도: $\mathcal{O}(N^2)$

J. 수열과 쿼리 2025

disjoint set, smaller to larger

예상 난이도 - Gold II ~ Platinum V

- ✓ 제출 9번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: -, -분
- ✓ 출제자: 이동빈^{leedongbin}, 박관우^{gwanwoopa}, 허용준^{sk14cj}

J. 수열과 쿼리 2025

- ✓ 1번 쿼리가 들어오면 수열에서 x 인 값을 모두 y 로 바꿔야합니다. 매 쿼리마다 일일이 다 찾을 순 없으니 같은 값은 하나의 집합으로 관리해야 합니다.
- ✓ A_i 와 A_i 집합의 대표 인덱스를 매핑하는 자료구조를 만들어줍니다. 수열의 최댓값이 30만이므로 길이가 30만인 배열을 사용하면 됩니다.
- ✓ 그리고 반복문을 돌면서 각 수열의 값 A_i 를 확인하면서 다음과 같이 처리합니다.
 - 만약 A_i 가 나온 적이 없다면 A_i 집합의 대표 인덱스는 i 가 됩니다.
 - 만약 A_i 가 나온 적이 있다면 A_i 집합에 i 를 추가합니다.

J. 수열과 쿼리 2025

- ✓ 1번 쿼리가 들어왔을 때 경우의 수를 다음과 같이 나눌 수 있습니다.
 - x 집합이 없는 경우
 - x 집합이 있고, y 집합이 없는 경우
 - x 집합이 있고, y 집합도 있는 경우

J. 수열과 쿼리 2025

- ✓ x 집합이 없다면 수행할 작업이 없습니다.
- ✓ x 집합이 있고, y 집합이 없다면 x 집합을 y 집합으로 바꾸면 됩니다. 이는 y 집합의 대표 인덱스를 x 집합의 대표 인덱스로 설정하고, x 집합의 대표 인덱스를 삭제합니다. 그리고 y 집합의 대표 인덱스의 수열의 값을 y 로 변경하면 됩니다.
- ✓ x 집합이 있고, y 집합도 있다면 x 집합을 y 집합에 합치면 됩니다. 이는 x 집합의 대표 인덱스를 y 집합에 추가하고, x 집합의 대표 인덱스를 삭제하면 됩니다.

J. 수열과 쿼리 2025

- ✓ 2번 쿼리가 들어오면 z 의 최상위 부모 p 를 찾아 A_p 를 출력하면 됩니다.

시간 복잡도 : $\mathcal{O}(M)$

그 외에도, 스몰 투 라지를 활용하여 문제를 해결할 수도 있습니다.

K. 터치 앤 리턴

dp, tsp, bitmask, knapsack, shortest path

예상 난이도 - **Platinum IV** ~ **Platinum III**

- ✓ 제출 1번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람: -, -분
- ✓ 출제자: 허용준^{sk14cj}

K. 터치 앤 리턴

- ✓ 몇 가지 관찰을 통하여, 이 문제에서 고려해야 할 통로의 수를 크게 줄일 수 있습니다.
- ✓ M 은 최대 300,000이지만, 지점의 수 N 이 최대 20이므로, 인접 행렬로 각 지점의 이동 소요 체력을 저장할 수 있으며, 실질적으로 고려해야 하는 통로의 수는 $n(n - 1)/2$ 개입니다.
- ✓ 위 과정에서 얻은 인접 행렬 w 에, 데이크스트라 혹은 플로이드-워셜 등 최단 경로 알고리즘을 사용하여, 각 지점 간의 최소 이동 소요 체력을 구하고 저장합니다.

K. 터치 앤 리턴

- ✓ 이전 페이지에서, 각 지점 간의 최소 이동 소요 체력을 구했기 때문에 비트마스크 DP를 사용하여, 중복되지 않은 지점 i 개를 방문하고 원점으로 돌아올 때 최소 소요 체력을 구하면 됩니다 ($1 \leq i \leq N$)
 - $tsp[mask][u]$: 지점 집합 $mask$ (항상 지점 1 포함)을 각 한 번씩 방문하여 지점 u 에 도달하는 최소 편도 비용.
 - 전이: $tsp[mask \cup \{v\}][v] = \min(tsp[mask \cup \{v\}][v], tsp[mask][u] + w[u][v])$
($v \notin mask$, 간선 (u, v) 존재)
 - 이렇게 얻은 각 $mask, u$ 에 대해 한 턴 왕복 비용은 $dp[mask][u] + w[1][u]$ 입니다.
- ✓ 그 외에도, 간선 가중치의 최대값이 낮으므로 다이얼 알고리즘 등을 사용하여 지점 i 개를 방문하고 원점으로 돌아올 때 최소 소요 체력을 구할 수도 있습니다.

K. 터치 앤 리턴

- ✓ 마지막으로, 체력 K 안에서 턴을 여러 번 수행할 수 있으므로 배낭 문제를 이용하여 최대 점수를 구할 수 있습니다.
- ✓ 점화식은 다음과 같습니다.
 - 방문 지점 t 개인 경우 최소 소모 체력 = C_t , 점수 = $(t - 1)^2$.
 - $dp[i] = \max(\max_{k \geq C_t} \{dp[i - C_t] + (t - 1)^2\}, dp[i - 1])$

시간복잡도: $\mathcal{O}(N^2 2^N)$