2025 ALOHA 단풍컵 (Div. 1)

Official Solutions



출제

- ✓ gggkik 한양대학교 컴퓨터소프트웨어학부 / ALOHA

검수

- ✓ daniel604
- ✓ fs_edge
- ✓ lkh3happy
- ✓ myyh1234
- ✓ tlsdydaud1

문제		의도한 난이도	출제자
A	두 번째로 큰 수	**Challenging**	gggkik
B	전기 회로 배치	**Medium**	gggkik
C	배열 뒤집기	**Challenging**	gggkik
D	XOR 계산 컴퓨터와 메모리	**Challenging**	gggkik
E	당도 높은 순열	**Medium**	gggkik
F	파티로 가는 길	**Hard**	gggkik
G	인버전 과제	**Medium**	gggkik
H	쉬운 순열 문제	**Hard**	gggkik, myyh1234



1A. 두 번째로 큰 수

`Sorting`, `Ad-hoc`
출제진 의도 – **Challenging**

- 처음 푼 팀: **아빠**, 41분
- 처음 푼 팀(Open Contest): **sadtreap**, 150분
- 출제자: `gggkik`

1A. 두 번째로 큰 수



- 카르테시안 트리를 O(N)에 구성합시다.
- 구간의 최솟값은 카르테시안 트리의 LCA와 같습니다.



1B. 전기 회로 배치

`Sorting`, `Ad-hoc`
출제진 의도 – **Medium**

- 처음 푼 팀: **N/A**, N/A분
- 처음 푼 팀(Open Contest): **platter**, 99분
- 출제자: `gggkik`



- a와 b를 이용하면 인접한 두 수의 위치를 바꿀 수 있습니다.
- 한 단계에서 한 원소의 위치가 최대 한번만 이동하도록 버블 정렬을 해야 합니다.
- 홀수 번째 단계에서는 홀수 번째 인덱스를 교환하고, 짝수 번째 단계에서는 짝수 번째 인덱스를 교환하면 항상 정렬이 가능합니다.



1C. 배열 뒤집기

`Ad-hoc`, `Constructive`, `Case-work`
출제진 의도 – **Challenging**

- 처음 푼 팀: **N/A**, N/A분
- 처음 푼 팀(Open Contest): **mickeyjung**, 149분
- 출제자: `gggkik`



- 첫 번째 열과 첫 번째 행을 모두 어떤 x로 만들어야 합니다.
- 첫 번째 열에서 더 적게 등장하는 수를 뒤집어서 첫 번째 열을 모두 같게 만듭시다.
- 첫 번째 행에서 더 적게 등장하는 수를 뒤집어서 첫 번째 행까지 모두 같게 만듭시다.
- 만약 남은 연산이 짝수 개라면, 한 위치에 전부 연산합니다. 홀수 개라면 연산들을 바뀐 위치를 알아볼 수 있는 여러분만의 패턴으로 기록합시다.



1D. XOR 계산 컴퓨터와 메모리

`Ad-hoc`, `Binary-search`
출제진 의도 – **Challenging**

- 처음 푼 팀: **N/A**, N/A분
- 처음 푼 팀(Open Contest): **N/A**, N/A분
- 출제자: `gggkik`



- -1이 없다고 가정하고 먼저 문제를 풀어봅시다.
- 우선 배열을 정렬한 상태로 step 2로 전송합니다.
- 트라이와 같은 원리로 최상위 비트별로 내려가며 탐색을 하면서 최대 XOR를 찾으면 됩니다.
- 최상위 비트부터 i번째 비트까지 모두 같은 수에 대해 $i-1$번째 비트는 정렬되어 있으므로 이분탐색으로 0에서 1이 되는 위치를 구할 수 있습니다.
- $N \times \log_2 X$의 메모리와 $\log_2 X \times \log_2 N$의 쿼리를 사용합니다.



- -1은 어떻게 복원할까요?
- 제곱근 분할법을 이용합시다.
- 우선 메모리 M을 $\sqrt{X} \times \sqrt{X}$ 크기의 정사각형으로 만듭니다.
- 각 열의 XOR과 각 행의 XOR을 저장한 길이 $2\sqrt{X}$의 배열을 추가로 저장하면 됩니다.
- 제한에 아주 타이트하게 통과할 수 있습니다.



1E. 당도 높은 순열

`Bfs, Construcive, Divide-and-Conquer`
출제진 의도 – **Medium**

- 처음 푼 팀: **안녕하세요**, 34분
- 처음 푼 팀(Open Contest): **yaksha**, 9분
- 출제자: `gggkik`



- 조건을 만족하는 여러 순열이 있습니다만, 출제 의도의 순열은 다음과 같습니다:
- 가장 중앙에 가장 큰 수를 배치합니다.
- 중앙을 기준으로 두 구역으로 나누고, 두 구역에 각각 사용하지 않은 수 중 큰 수 순서대로 배치합니다.
- 이러한 과정을 반복하여 답을 구성한다면, 당도 1.6×10^{15} 의 순열을 얻을 수 있습니다.



1F. 파티로 가는 길

`Ad-hoc`, `Bfs`

출제진 의도 – **Hard**

- 처음 푼 팀: **안녕하세요**, 134분
- 처음 푼 팀(Open Contest): **yaksha**, 53분
- 출제자: `gggkik`



- 위치 A에서 B까지 격자상에서 경로를 관찰해봅시다.
- UDLR 4개의 문자로 표현해야 할 것 같지만, 사실 왔던 길로 절대 돌아가지 않으므로 좌회전, 우회전, 직진으로 표현한 3개의 문자를 사용하는 길이 D의 문자열로 표현할 수 있습니다.
- 깊이를 D로 가지는 트리를 만들어 각 높이마다 차수를 통해 문자를 표현합시다.



1G. 인버전 과제

Ad-hoc
출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **일리만 은디아예**, 22분
- ✓ 처음 푼 팀(Open Contest): **mickeyjung**, 10분
- ✓ 출제자: `gggkik`



- 앞에 수부터 하나씩 0으로 만들어 봅시다.
- 우선 $[1, 1]$을 뒤집었을 때 인버전이 늘어났다면 A_1은 처음에 0이었다는 사실을 알 수 있습니다.
- 인버전이 늘어났다면 다시 $[1, 1]$을 뒤집어주고 $[2, 2]$에서 같은 알고리즘을 반복합시다.
- 연산을 $2N$회 사용하여 인버전을 0으로 만들 수 있는 알고리즘을 완성했습니다.



- i번째 위치를 뒤집었을때 인버전이 늘어났다면, 바로 $[i, i]$로 다시 뒤집어주는 대신 $[i+1, i+1]$를 탐색 할 때 $[i, i+1]$로 이전 위치를 뒤집어주면서 동시에 탐색할 수 있습니다.
- 쿼리를 최대 $N-1$번 사용하여 인버전을 0으로 만들 수 있습니다.
- 참고로, 벚꽃컵 기출인 "벚꽃과 단풍"과 유사한 문제입니다.



1H. 쉬운 순열 문제

Ad-hoc

출제진 의도 – **Hard**

- 처음 푼 팀: **N/A**, N/A분
- 처음 푼 팀(Open Contest): **sadtreap**, 29분
- 출제자: `gggkik`, `myyh1234`



- ✓ (x, y, z)에 대해 $x \neq z$이므로, x와 z에서 다른 비트 하나가 무조건 있습니다.
- ✓ 모든 i에 대해 i번째 비트가 다른 x, z가 등장하는 순서쌍을 만들어줍시다.
- ✓ Case Work를 통해 몇개의 순열을 만들어 전부 넣어줍니다.
- ✓ i번째 비트가 1인 집합을 A, 0인 집합을 B라고 하고 S'을 S의 순서를 뒤집은 수열이라고 합시다.
- ✓ $A + B, A' + B, B' + A, B' + A'$를 전부 넣어주면 i번째 비트가 다른 모든 순서쌍 (x, y, z)을 커버할 수 있습니다.
- ✓ 참 쉽죠?