



## 230622 모의대회 해설

AIKon 2023

# 연습대회 결과

## Div. 3

**All Solved** 🏆

이동훈 donghoony

**Solved 3**

이동훈 donghoony

**Solved 2**

이동훈 donghoony

**Solved 1**

이동훈 donghoony

## Div. 2

**All Solved** 🏆

이동훈 donghoony

**Solved 3**

이동훈 donghoony

**Solved 2**

이동훈 donghoony

이동훈 donghoony

이동훈 donghoony

## Div. 1

**All Solved** 🏆

이동훈 donghoony

**Solved 4**

이동훈 donghoony

**Solved 2**

이동훈 donghoony

이동훈 donghoony

이동훈 donghoony

이동훈 donghoony

이동훈 donghoony

# Div. 3

번호	제목	난이도
3A	Pen Pineapple Apple Pen	Easy
3B	증가 배열 만들기	Easy
3C	카우버거	Medium
3D	두찌 수열	Medium
3E	서강근육맨	Hard

## 3A Pen Pineapple Apple Pen

string, implementation

- ✓ 주어진 문자열에서 pPAp가 몇 번 등장하는지 세어야 합니다.
- ✓ 앞에서부터 차례로 확인하면서 해당 문자열인지 확인합니다.
- ✓ 하나의 펜은 하나의 pPAp만 속해야 하므로, 인덱스 조절을 신경써야 합니다.

## 3B 증가 배열 만들기

ad\_hoc, constructive

- ✓ 항상 오른쪽이나 아래로만 이동할 수 있으므로, 중요한 성질을 발견할 수 있습니다.
- ✓ 배열에서 행 번호 + 열 번호인  $i + j$  의 값이 같은 칸끼리 서로 묶어서 1부터 번호를 부여합니다.
- ✓  $i, j$ 는 1부터 번호매김,  $(1, 1)$ 을 1로,  $(i, j)$ 를  $i + j$ 로 번호를 부여합니다.
- ✓ 이 때, 가장 오른쪽 아래 칸의 번호보다  $K$ 가 작다면, 배열을 만들 수 없습니다.
- ✓ 그 외에는 항상 위에서 설명한 방법으로 구성할 수 있습니다.

## 3C 카우버거

sorting

- ✓ 버거 세트를 만들되, 세트를 만든 뒤 모든 가격의 합이 최소가 되어야 합니다.
- ✓ 10퍼센트 할인이므로, 가격이 비쌀수록 더 많은 할인을 받을 수 있습니다.
- ✓ 할인받을 수 있는 개수는  $\min(B, C, D)$ 입니다.
- ✓ 위 인덱스보다 앞에 있는 메뉴는 할인이 가능하므로 0.9를 곱합니다.
- ✓ 그렇지 않은 경우는 어떻게 해도 할인받지 못하는 메뉴이므로, 그냥 더하면 됩니다.

## 3D 두께 수열

implementation, simulation

- ✓ 문제를 잘 읽어보면 쉽게 해결할 수 있습니다!
- ✓ 루프를 만드는데 필요한 최대 단계수는 1,000을 넘지 않는다고 쓰여 있습니다.
- ✓ 직접 주어진 배열을 연산하고, 이후 0으로만 이루어져있는지 확인하는 과정을 1000번 반복합니다.
- ✓ 만약 이후에도 0이 아닌 수가 존재한다면, 루프가 존재합니다.

## 3E 서강근육맨

sorting

- ✓ 두 개의 합들의 최댓값을 최소화해야 합니다.
- ✓ 거꾸로 생각해 봅시다. 배열에서 가장 큰 값은 누구와 짝을 지어야 할까요?
- ✓ 만약 배열의 길이가 홀수라면, 가장 큰 값은 짝을 짓지 않는것이 이득입니다.
- ✓ 짝수라면, 가장 작은 값과 짝을 지어야 합니다.
- ✓ 짝지은 두 원소를 제외한 나머지에 대해서도 같은 방식으로 짝지어줄 수 있습니다.
- ✓ 배열을 정렬하는 데  $O(N \log N)$ , 서로 짝지어주는 데  $O(N)$ 입니다.



# Div. 2

번호	제목	난이도
2A	오리	Easy
2B	부당한 퍼즐	Medium
2C	빛의 돌 옮기기	Medium
2D	Gazzzua	Medium
2E	Swap the elements	Hard

## 2A 오리

string

- ✓ 오리가 총 몇 마리인지 구해야 합니다. 가능한 오리의 수를 최소화해야 합니다.
- ✓ 앞에서부터 훑으면서, **quack**이 연속으로 있는 구간을 계속해서 지워나가는 방법이 통과합니다.
- ✓ **quac**와 같은 경우는 오리가 없는 경우입니다.
- ✓ 주어진 문자열을 계속해서 지워나가면서, 최종 오리의 마릿수를 구합니다.

## 2B 부당한 퍼즐

ad\_hoc

- ✓ 문제에서 사용할 수 있는 두 가지 연산에서, 변하지 않는 것을 찾아봅시다.
- ✓ 한 원소를 기준으로, 그 원소와 이웃한 좌/우 원소는 바뀌지 않습니다.
- ✓ 해당 아이디어를 기반으로 간단하게 구현할 수 있습니다.

## 2C 빛의 돌 옮기기

dp

- ✓ 빛의 돌을 끌고 가거나, 들고 가거나 둘 중 하나를 선택할 수 있습니다.
- ✓ 전략을 바꿀 때마다  $K$ 의 추가 비용이 발생합니다.
- ✓ 간단한 바텀업 DP를 생각해봅시다.  $i$ 번째 날의 끌고가는 비용은
- ✓  $\min(\text{전날 끌고가는 비용} + \text{오늘 비용}, \text{전날 들고가는 비용} + \text{오늘 비용} + K)$ 입니다.
- ✓ 같은 방법으로 들고가는 비용도 구할 수 있습니다.

## 2D Gazzzua

greedy

- ✓ 왼쪽부터 진행하면서 내리막이 발생하면 그때 팔아버리는 그리디 전략이 먼저 떠오를 수 있습니다.
- ✓ 1, 2, 1, 2, 100 과 같은 경우, 4일 내내 산 다음에 마지막 날에 파는게 이득입니다.
- ✓ 다른 전략으로, 팔 날을 미리 구해 봅시다.
- ✓ 코인을 팔 날을 정하기 위해서는 다른 날들보다 더 큰 날이어야 합니다.
- ✓ 배열을 거꾸로 순회하면 정답을 구할 수 있습니다. 아이디어가 어려운 문제입니다.

## 2E Swap the elements

ad\_hoc, constructive

- ✓ 우선 만들 수 없는 경우부터 생각해 봅시다. 비둘기집 원리를 통해 알 수 있습니다.
- ✓ 가장 많이 등장한 원소의 등장 횟수가  $\frac{N}{2}$  보다 크면 배열을 구성하는 것이 불가능합니다.
- ✓ 그렇지 않은 경우에는 반드시 만들 수 있습니다. 모든 원소의 등장 횟수가  $\frac{N}{2}$  이하임에 주목합시다.
- ✓ 방법 중 하나로, 주어진 배열을  $(a_i, i)$ 를 쌍으로 하는 배열을  $a_i$ 를 기준으로 정렬합시다.
- ✓ 정렬된 배열은 같은 값이 연속돼서 나타납니다.
- ✓  $i$ 번째 원소와  $i + \lfloor \frac{N}{2} \rfloor$ 번째 원소를 바꿔줍니다. 모든 원소의 등장 횟수의 성질에 따라, 두 원소는 반드시 다릅니다.

# Div. 1

번호	제목	난이도
1A	내일 할거야	Medium
1B	넴모넴모 2020	Medium
1C	무술 연습	Hard
1D	차이를 최대로 2	Hard
1E	트리 만들기	Challenging

# 1A 내일 할거야

greedy, sorting

- ✓ 최대화해야하는 것은 첫 날로부터 **연속**으로 쉴 수 있는 날입니다.
- ✓ 간단하게 데드라인이 가까운 것부터 처리하는 그리디 전략이 통과합니다. 왜 그럴까요?
- ✓ 노는 날을  $K$ 만큼 확보한다고 하면, 다른 일들의 데드라인이  $K$ 만큼 줄어드는 것과 같습니다.
- ✓ 이 때, 줄어든 일끼리의 데드라인 대소관계는 불변하므로, 같은 전략을 사용해도 됩니다.



# 1B 메모메모 2020

binary\_search

- ✓ 오른쪽으로 뺀어나가는 레이저와 위로 뺀어나가는 레이저를 독립적으로 생각해 봅시다.
- ✓ 우선, 오른쪽으로 뺀어나가는 레이저는 쉽게 구할 수 있습니다.
- ✓ 배열의 값에서 해당 좌표를 빼 주면 됩니다.
- ✓ 위로 뺀어나가는 레이저의 경우에는 조금 생각해봐야 합니다.
- ✓ 문제에서 제시된 조건에 주목합시다. ( $1 \leq y \leq N - 1$ 에 대해  $a_y \geq a_{y+1}$ )
- ✓ 제시된 조건에 따라, 위로 올라가다가 해당 값보다 작아지는 최초의 인덱스를 구하면 됩니다.
- ✓  $a_i$ 가 매우 클 수 있으므로, 이분 탐색으로 해당 값을 구합시다.
- ✓ 만약 독립적으로 구했다면, 둘 다 없앨 수 있는 경우에는 겹치는 칸을 빼 주어야 합니다.

## 1C 무술 연습

graphs, topological\_sorting

- ✓ 바라보는 사람을 향해 방향성이 있는 간선을 그려 봅시다.
- ✓ 최초에 아무도 가리키지 않은 사람부터 활을 잡을 수 있습니다.
- ✓ 이후에 활을 잡은 사람이 가리킨 사람은 반드시 방패를 들어야 합니다.
- ✓ 위상 정렬과 같은 방식으로 해결할 수 있지만, 사이클이 존재할 수 있습니다.
- ✓ 사이클에 속하는 한 사람에게 활을 쥐어 줌으로써 해결할 수 있습니다.

# 1D 차이를 최대로 2

greedy, sorting



# 1E 트리 만들기

tree, priority\_queue

