INU 코드페스티벌 2024 풀이

Official Solutions

2024년 9월 26일

문제		의도한 난이도	출제자
A	양파 실험	**Easy**	김영현 kimyh9797
B	Rook	**Easy**	허용준 sk14cj
C	샷건	**Easy**	이동빈 leedongbin
D	마법의 나침반	**Medium**	김영현 kimyh9797
E	장난감 자물쇠	**Medium**	허용준 sk14cj
F	선물 고르기	**Medium**	이동빈 leedongbin
G	불꽃놀이	**Medium**	김영현 kimyh9797
H	Draw A Perfect Circle	**Medium**	이동빈 leedongbin
I	풍선 터트리기	**Hard**	김영현 kimyh9797
J	횃불이 키우기	**Hard**	허용준 sk14cj
K	수 만들기	**Hard**	이동빈 leedongbin

A. 양파 실험

`simulation`
출제진 의도 – **Easy**

- 제출 117번, 정답 71명 (정답률 60.684%)
- 처음 푼 사람: **나는몇등일까**, 1분
- 출제자: 김영현`kimyh9797`

A. 양파 실험

- 주어진 조건을 그대로 구현하는 문제입니다.
- 칭찬 양파의 길이를 X, 비난 양파의 길이를 Y 라고 합니다. X, Y 의 초기값은 1 입니다.
- 아래 과정을 N 일 동안 반복합니다.
  - 1. X 에 A 를 더하고, Y 에 B 를 더합니다.
  - 2. $X < Y$ 인 경우 X 와 Y 의 값을 바꿉니다.
  - 3. $X = Y$ 인 경우 Y 를 1 만큼 감소시킵니다.

시간복잡도 : $\mathcal{O}(N)$

B. Rook

ad-hoc

출제진 의도 – **Easy**

- 제출 308번, 정답 50명 (정답률 16.234%)
- 처음 푼 사람: **CH2MI**, 6분
- 출제자: 허용준sk14cj

B. Rook

- Pawn A가 $(0, n)$에 위치할 때 (n은 1 이상의 정수)
  - Pawn B가 $(0, m)$이고 m이 n보다 작을 경우 B를 우회하여 A를 잡아야 하므로 3, 그 외의 경우는 1입니다.

- Pawn A가 $(n, 0)$에 위치할 때 (n은 1 이상의 정수)
  - 위와 같은 이유로 Pawn B가 $(m, 0)$이고 m이 n보다 작을 경우 3, 그 외의 경우는 1입니다.

- Pawn A가 (n, m)에 위치할 때 (n, m 모두 1 이상의 정수)
  - Pawn B가 어디에 위치하든 R은 2번의 이동으로 A를 잡을 수 있습니다.

시간복잡도 : $\mathcal{O}(1)$

C. 샷건

`implementation`, `brute force`
출제진 의도 – **Easy**

- 제출 73번, 정답 53명 (정답률 72.603%)
- 처음 푼 사람: **나는몇등일까**, 13분
- 출제자: 이동빈 `leedongbin`

C. 샷건

- ✓ 키보드 배열의 크기가 충분히 작으므로, 모든 경우에 대해 시도해볼 수 있습니다.
- ✓ 임의의 키를 골라 상하 좌우 대각선을 포함한 9개의 키를 알아내고, 무작위 순서로 입력된 문자열과 같은지 비교해보면 됩니다.
- ✓ 다음과 같은 여러 가지 방법으로 구현할 수 있습니다.
  - 크기 127 이상의 배열을 만들어, 각 9개 문자의 ASCII값에 해당하는 인덱스에 개수를 저장한 후 개수가 같은지 비교합니다.
  - 입력된 문자열과 알아낸 문자열을 정렬하여, 서로 같은지 비교합니다.
  - 입력된 문자열의 모든 문자가 임의로 고른 키의 상하 좌우 대각선에 포함되는지 삼중 for 문으로 확인합니다.
- ✓ 시간복잡도는 키보드의 행의 길이를 R, 열의 길이를 C라고 할 때 $\mathcal{O}(RC)$가 됩니다.

시간복잡도 : $\mathcal{O}(RC)$

D. 마법의 나침반

`implementation`, `bruteforce`
출제진 의도 – **Medium**

- 제출 122번, 정답 43명 (정답률 36.066%)
- 처음 푼 사람: **엄준식**, 24분
- 출제자: 김영현`kimyh9797`

D. 마법의 나침반

- 나침반 사용 기록마다 보물이 존재할 수 있는 곳을 좁혀나갈 수 있습니다.
- 가능한 위치가 한 곳으로 좁혀지면 해당 위치가 보물이 존재하는 곳입니다.
  - 반드시 단 하나의 정답이 존재하기 때문입니다.
- N 이 충분히 작기 때문에 각각 위치에서 모든 나침반 사용기록을 만족하는지 확인하는 방법으로도 문제를 해결할 수 있습니다.
  - 모든 위치에서 나침반 사용 기록이 일치하는 횟수를 계산합니다.
  - 일치하는 횟수가 M인 곳이 보물의 위치입니다.

시간복잡도 : $\mathcal{O}\left(N^2M\right)$

E. 장난감 자물쇠

`math`, `sort`
출제진 의도 – **Medium**

- 제출 246번, 정답 22명 (정답률 8.943%)
- 처음 푼 사람: **cu**, 37분
- 출제자: 허용준 `sk14cj`

E. 장난감 자물쇠

- 정해
- 수열 A의 원소값은 0부터 $N-1$까지 각각 하나씩 존재하고, 이를 순서대로 나열해야 하므로 자물쇠의 잠금을 해제하려면 $A[i] = i (0 \leq i \leq N-1)$로 만들 수 있어야 합니다.
- 문제에 제시된 규칙에 따라, $A[i]$는 $A[i+kx]$로 자유롭게 이동할 수 있습니다. (x는 정수, $0 \leq i+kx \leq N-1$)
- 따라서 0부터 $N-1$까지의 모든 정수 i가 $(A[i]-i) \mod k = 0$을 만족하면 Yes를 출력하고, 그렇지 않다면 No를 출력하면 됩니다.

시간복잡도 : $\mathcal{O}(N)$

E. 장난감 자물쇠

- 다른 풀이
- 인덱스를 k로 모듈러 연산한 값이 같은 원소끼리 정렬을 합니다.
- 정렬을 마친 이후, $0 \leq i \leq N-1$에서 $A[i] \neq i$가 하나라도 존재한다면 No를 출력하고, 그렇지 않다면 Yes를 출력합니다.
- 정렬을 할 때, 시간복잡도가 $\mathcal{O}(NlogN)$인 정렬을 사용하면 제한 시간 내에 통과하지만, $N \leq 200\,000$이므로 버블 정렬 등 $\mathcal{O}(N^2)$인 정렬을 사용할 경우 시간초과가 발생합니다.

시간복잡도 : $\mathcal{O}(NlogN)$

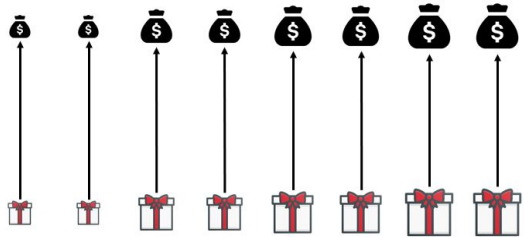
F. 선물 고르기

`greedy`, `sort`, `data structure`
출제진 의도 – **Medium**

- 제출 54번, 정답 11명 (정답률 20.370%)
- 처음 푼 사람: **동동구리**, 85분
- 출제자: 이동빈 `leedongbin`

F. 선물 고르기

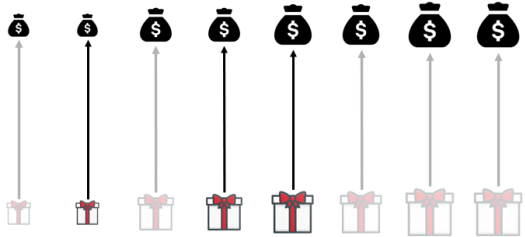
- 먼저, 선물과 선물 상자를 각각 크기순으로 정렬해봅시다.



- 선물 상자에 모든 선물을 담을 수 없는 경우는 주어지지 않는다는 조건에 의해, 정렬된 상태에서는 선물과 선물 상자를 항상 그림과 같이 일대일 대응시킬 수 있습니다.

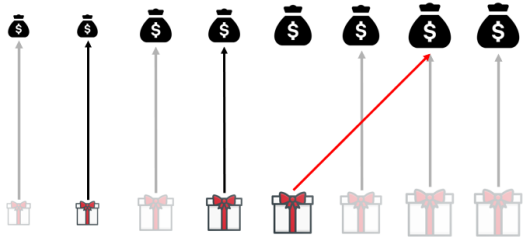
F. 선물 고르기

- 이제 앞의 K 명이 선물 상자를 골라 가져갔다고 가정해봅시다.



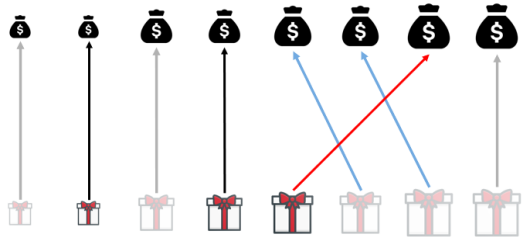
F. 선물 고르기

- 일단, 고른 상자에 담을 수 있는 가장 큰 선물을 담아봅시다.



- 이제 일대일 대응관계를 만족하도록 화살표를 적절히 옮겨야 합니다.

F. 선물 고르기



- 그림과 같이 옮겨진 파란색 화살표들은 항상 기존보다 작거나 같은 선물을 가리키게 되므로, 각 선물 상자에는 선물 상자의 크기보다 작거나 같은 선물이 들어있다는 문제 조건을 여전히 만족합니다.
- 따라서 남아있는 가장 큰 상자에 담을 수 있는 가장 큰 선물의 크기가 정답이 됩니다.
- 이는 map이나 multiset 등 다양한 자료구조를 이용하여 구현할 수 있습니다.

시간복잡도 : $\mathcal{O}(N \log N)$

G. 불꽃놀이

`greedy`, `sort`, `two pointer`
출제진 의도 – **Medium**

- 제출 11번, 정답 1명 (정답률 9.091%)
- 처음 푼 사람: **내위에서내려가**, 142분
- 출제자: 김영현 `kimyh9797`

G. 불꽃놀이

- 문제를 2단계로 나누어 생각해봅시다.
  - 1. 원소가 1개 혹은 2개로 이루어진 화려한 정도가 K 이상인 조합을 찾아냅니다.
  - 2. 1번에서 찾아낸 조합을 순서에 맞게 배치합니다.
- 두번째 단계는 **비내림차순**으로 배치하면 항상 만든 모든 조합을 조건에 맞게 배치할 수 있습니다.
- 즉, 첫번째 단계만 고려하여 화려한 정도가 K 이상을 조합을 가장 많이 만드는 것이 이 문제의 핵심입니다.

G. 불꽃놀이

- K 이상의 화려한 정도를 가진 폭죽은 원소가 1 개인 조합으로 생성합니다.
  - 각각의 폭죽의 화려한 정도가 이미 K 이상이므로, 다른 폭죽과 같이 사용할 필요는 없습니다.
- K 미만의 화려한 정도를 가진 폭죽은 2 개를 적절히 조합하여 K 이상의 조합을 만들어야 합니다.
  - 이는 화려한 정도 순으로 정렬하고, 투 포인터를 활용하면 효과적으로 찾을 수 있습니다.

시간복잡도 : $\mathcal{O}\left(N \log N\right)$

H. Draw A Perfect Circle

`geometry`, `two_pointer`
출제진 의도 – **Medium**

- 제출 40번, 정답 6명 (정답률 17.500%)
- 처음 푼 사람: **cu**, 60분
- 출제자: 이동빈`leedongbin`

H. Draw A Perfect Circle

- 정확도는 점들의 원점으로부터의 거리에 의해 계산되므로, 각 점의 좌표를 원점으로부터의 거리로 바꿔줍시다.
- 점들을 원점으로부터의 거리의 오름차순으로 정렬하여,
  각 점을 $p_1, p_2, \cdots p_N$, 원점으로부터의 거리를 $d_1, d_2, \cdots d_N$ 이라고 합시다.

H. Draw A Perfect Circle

- 너비 K 인 구간에, 원점으로부터 가장 가까운 점인 p_1 을 포함한다고 생각해봅시다.
- 구간 $[0 \sim d_1)$ 에는 어떤 점도 존재하지 않으므로, $[d_1 \sim d_1 + K]$ 인 구간을 골라야 가장 많은 점을 포함할 수 있습니다. (즉, $R = d_1 + \frac{K}{2}$ 일 때 최적입니다.)
- 이때 $d_x \leq d_1 + K$ 를 만족하는 가장 큰 x 를 X 라 하면, 구간에 포함되는 점의 개수는 X 개 입니다.

H. Draw A Perfect Circle

- 이제 점 p_1 을 포기하고 p_2 부터 구간에 포함한다고 생각해봅시다.
- 마찬가지로 구간 $[0 \sim d_2)$ 에는 p_1 이외의 점이 존재하지 않으므로,
  $[d_2 \sim d_2 + K]$ 인 구간을 고를 때 최적이 됩니다.
- 이 과정을 일반화하면,
  - 점 $p_{i-1} (2 \leq i \leq N)$ 을 포기하고 p_i 부터 구간에 포함한다고 가정하면 구간이 $[d_i \sim d_i + K]$ 일 때 최적이 되며,
  - $d_y \leq d_i + K$ 를 만족하는 가장 큰 y 를 Y 라 하면 구간에 포함되는 점의 개수는 $Y - (i - 1)$ 개가 됩니다.
- 이제 구간을 투 포인터로 관리하면서, 가장 많은 점을 포함하는 구간을 구하면 문제를 해결할 수 있습니다.

H. Draw A Perfect Circle

- 그런데 이전 슬라이드에서 언급한 $d_y \leq d_i + K$ 를 실수 범위에서 계산하면 부동소수점 오차에 의해 오답을 받게 될 수 있으므로, 다음과 같은 방법으로 정수 범위에서 계산해주어야 합니다.
- 오름차순으로 정렬된 원점으로부터의 거리의 **제곱** 배열을 $D_1, D_2, \cdots D_N$ 이라고 합시다.
- 이제 $\sqrt{D_y} \leq \sqrt{D_i} + K$ 를 계산하면 됩니다.
- 양변을 제곱하면 $D_y \leq D_i + K^2 + 2K\sqrt{D_i}$ 가 됩니다.
- $D_i + K^2$ 를 좌변으로 넘기면 $D_y - D_i - K^2 \leq 2K\sqrt{D_i}$ 가 되고,
- 다시 양변을 제곱하면 $(D_y - D_i - K^2)^2 \leq 4K^2 D_i$ 가 됩니다. (이때, $D_y - D_i - K^2 < 0$ 인지 먼저 체크해주어야 합니다.)
- 이제 부등식의 모든 항이 정수이므로, 정수 범위에서 계산할 수 있습니다.

시간복잡도 : $\mathcal{O}(N \log N)$

I. 풍선 터트리기

`binary search`
출제진 의도 – **Hard**

- 제출 97번, 정답 2명 (정답률 2.062%)
- 처음 푼 사람: **cu**, 110분
- 출제자: 김영현`kimyh9797`

- 게임이 가장 늦게 끝나는 경우를 생각해봅시다.
- N, x, y, z 가 모두 10^9 인 경우가 가장 긴 게임입니다.
- 이 게임은 대략 $\frac{10^{18}}{3}$ 분 정도의 시간이 필요합니다.
- 게임이 끝나는 시간은 항상 64 비트 정수형으로 표현할 수 있습니다.

- 게임이 끝나는 시간을 T 라고 생각해봅시다.
- A 는 $x, 2x, 3x...$ 분마다 풍선을 터트립니다.
- 그러므로 T 분 동안 A 는 $\lfloor \frac{T}{x} \rfloor$ 개의 풍선을 터트립니다.
- 같은 방법으로 B 는 $\lfloor \frac{T}{y} \rfloor$ 개의 풍선을, C 는 $\lfloor \frac{T}{z} \rfloor$ 개의 풍선을 터트립니다.
- 이분탐색을 이용하면 게임이 끝나는 시간 T 를 찾을 수 있습니다.

- ✓ $\lfloor \frac{T}{x} \rfloor + \lfloor \frac{T}{y} \rfloor + \lfloor \frac{T}{z} \rfloor$ 의 결과가 N 이상인 가장 작은 T 를 찾습니다.
- ✓ 이렇게 찾은 $\lfloor \frac{T}{x} \rfloor + \lfloor \frac{T}{y} \rfloor + \lfloor \frac{T}{z} \rfloor$ 의 결과는 $N, N+1, N+2$ 중 하나입니다.
  - 풍선이 터지는 시점은 겹칠 수 있습니다.
  - 최대 3 개의 풍선이 터지는 시점이 겹칠 수 있으므로 다음의 결과가 나오게 됩니다.

I. 풍선 터트리기

- T 분에 풍선을 터트리는 사람은 다음과 같이 알 수 있습니다.
  - $T \mod x$가 0인 경우 A는 T 분에 풍선을 터트립니다.
  - $T \mod y$가 0인 경우 B는 T 분에 풍선을 터트립니다.
  - $T \mod z$가 0인 경우 C는 T 분에 풍선을 터트립니다.
- 이를 종합해보면 $\lfloor \frac{T}{x} \rfloor + \lfloor \frac{T}{y} \rfloor + \lfloor \frac{T}{z} \rfloor$의 결과가 N이면 T 분에 풍선을 터트리는 사람 중 마지막으로 풍선을 터트리는 N번째 풍선을 터트립니다.
- $N+1$이라면 마지막에서 2번째, $N+2$라면 마지막에서 3번째 사람이 N번째 풍선을 터트립니다.

시간복잡도 : $\mathcal{O}(\log N)$

J. 횃불이 키우기

`math, dp`
출제진 의도 – **Hard**

- 제출 9번, 정답 0명 (정답률 0%)
- 처음 푼 사람: -, -분
- 출제자: 허용준 `sk14cj`

J. 횃불이 키우기

- 이 문제를 해결하기 위해서는 수학적인 관찰이 필요합니다.

- 2^{37} 은 $137\,438\,953\,472$ 입니다. 그리고, A_i 의 최솟값은 -500 이고, N 의 최댓값은 $200\,000$ 이니 계속 먹이를 섭취한다고 가정해도 어림잡아 1 억 이상 줄일 수 없습니다.

- 실제로 k 가 37 일 때, 최선을 다해 횃불이를 키우는 경우 중 크기가 가장 작을 때를 나타내는 데이터는 N 이 $200\,000$, s 가 1, A_i 가 -500 $(1 \leq i \leq N)$ 인 경우입니다. 이 데이터에서 37 일 차 횃불이의 최대 크기는 $137\,338\,971\,972$ 라는 값이 나오고, 10^{11} 보다 큽니다. 따라서, k 가 36 보다 클 때, 무조건 Mega 를 출력하면 됩니다.

  k 가 36 보다 큰 경우 시간복잡도 : $\mathcal{O}(1)$

J. 횃불이 키우기

- 이제 k 가 36 보다 같거나 작은 경우를 생각해 봅시다.

- $(N+1) \times (k+1)$ 크기의 다이나믹 프로그래밍을 위한 배열 dp 을 생성하고, i 일차에 j 번 강화했을 때 횃불이의 최대 크기를 $dp[i][j]$ 에 저장하도록 점화식을 세워서 문제를 풀면 됩니다.

- 점화식은 다음과 같습니다.

- $$dp[i][j] = \begin{cases} dp[i-1][j] + A_i, \ if \ j = 0 \\ dp[i][j] = \max(dp[i-1][j] + A_i, dp[i-1][j-1] \times 2), \ if \ 1 \le j \le k \end{cases}$$

- 단, 위 점화식에서 dp 테이블 원소의 값이 0 이하인 경우에는 연산을 생략합니다.

k 가 36 보다 작거나 같을 경우 시간복잡도 : $\mathcal{O}(Nk)$

K. 수 만들기

`bfs`

출제진 의도 – **Hard**

- 제출 1번, 정답 0명 (정답률 0%)
- 처음 푼 사람: -, -분
- 출제자: 이동빈`leedongbin`

K. 수 만들기

- 이 문제는 수와 연산을 그래프로 모델링하여 해결할 수 있습니다.
- $[1 \sim 5\,000\,000)$ 의 모든 수를 그래프의 정점이라고 생각해봅시다.
- 그러면 수 X 에 C 의 비용을 들여 수 Y 로 만드는 행동은, 정점 X 에서 Y 로 가는 비용 C 인 간선이 됩니다.
- 이제 이렇게 만든 그래프에서, A 에서 B 로 가는 최단경로를 구하면 됩니다.

K. 수 만들기

- 각 수에서 2번 연산에 의한 간선은 무한하고, 1번 연산에 의한 간선의 개수는 $(\text{자릿수} \times 10 - 1)$ 개입니다.
- 간선마다 비용이 서로 다르므로 데이크스트라 등의 최단경로 알고리즘을 사용해야 하지만, 간선과 정점이 너무 많아 시간초과를 받게 될 것입니다.
- 몇 가지 관찰을 통해, 그래프에서 불필요한 간선을 제거해봅시다.

K. 수 만들기

- 1번 연산에서, 비용이 $k(k > 1)$인 연산은 비용이 1인 연산을 k번 반복하면 같은 결과를 만들 수 있습니다.
- 마찬가지로, 2번 연산에서도 비용이 $y(y > 1)$인 연산은 비용이 1인 연산을 y번 반복하면 같은 결과를 만들 수 있습니다.
- 또한, 2번 연산에 의해 수가 $5\,000\,000$을 초과하는 경우는 고려하지 않아도 됩니다.
  - 만약 $4\,000\,001 \to 5\,000\,001 \to 4\,999\,999$와 같이 $5\,000\,000$을 초과해야 하는 경우, $4\,000\,001 \to 3\,999\,999 \to 4\,999\,999$와 같이 2번 연산을 먼저 수행하면 $5\,000\,000$을 초과하지 않는 경로가 존재합니다.

K. 수 만들기

- 이제 모든 간선의 비용이 1이므로, bfs를 이용하여 문제를 해결할 수 있습니다.
- 수 X 에서 1번 연산에 의한 간선의 개수는 $\log_{10} X$ 에 비례하고, 2번 연산에 의한 간선의 개수는 2개 또는 1개입니다.
- 입력 제한의 상한을 K 라고 하면, 정점의 수 V 는 K 에 비례하고, 총 간선의 수 E 는 $K \log K$ 에 비례합니다.
- 따라서 시간복잡도는 $\mathcal{O}(V + E) = \mathcal{O}(K \log K)$ 가 되고, 상수가 충분히 작으므로 주어진 시간 안에 동작합니다.

시간복잡도 : $\mathcal{O}(K \log K)$