Sogang ICPC Team

2025 서강대학교 K512컵

에디토리얼

주최 Sogang ICPC Team

후원 서강대학교 SOGANG UNIVERSITY STARTLINK

K512 CUP

A	3대 512	**Easy**	박민성minstar
B	전철 통학	**Easy**	박성준greedev
C	사막 탐험	**Easy**	채성우lem0nad3
D	지정좌석제	**Easy**	이재찬dlwocks1209
E	Magical Trees	**Medium**	채성우lem0nad3
F	취향 변화	**Medium**	강효규djs100201
G	LCS Making	**Medium**	채성우lem0nad3
H	점진적인 수열	**Medium**	강효규djs100201
I	광부가 될 수 있다면	**Medium**	이재찬dlwocks1209
J	거의 같은 문자열	**Hard**	강효규djs100201
K	Ultimate A + B	**Challenging**	이재찬dlwocks1029



A. 3대 512

`implementation` 출제진 의도 – **Easy**

- 제출 109번, 정답 53명 (정답률 48.624%)
- 처음 푼 사람: **raararaara**, 1분
- 출제자: 박민성 `minstar`

A. 3대 512



- 전체 학생들의 $Ai + Bi + Ci$를 탐색하면서 총합이 512 이상인 값 중 최솟값을 찾으면 됩니다.
- 시간복잡도: $O(N)$



B. 전철통학

출제진 의도 – **Easy**

- 제출 157번, 정답 47명 (정답률 29.936%)
- 처음 푼 사람: **말년일병**, 5분
- 출제자: 박성준 `greedev`

B. 전철통학



- 임의의 역에 대해, 역의 배차 간격을 T 분, 이 역에 민수가 도착하는 데 걸리는 시간을 K 분이라 합시다.
- 민수가 이 역에서 전철을 가장 빠르게 타는 시간은 T 의 배수들 중 K 이상인 가장 작은 수입니다.
- 세 역에 대해 각각 가장 빠르게 타는 시간을 구하고, 그 중 가장 빠른 시간을 출력합니다.
- 가장 빠르게 타는 시간은 $\left\lfloor \frac{K+T-1}{T} \right\rfloor \times T$ 로 쉽게 구할 수 있습니다.



C. 사막 탐험

`geometry` 출제진 의도 – **Easy**

- 제출 126번, 정답 37명 (정답률 29.365%)
- 처음 푼 사람: **raararaara**, 12분
- 출제자: 채성우 `lem0nad3`

C. 사막 탐험



✓ 다음 두 경우 중 최솟값이 답입니다.

1. 오아시스를 지나지 않는 경우
2. 오아시스를 지나는 경우

C. 사막 탐험



1. 오아시스를 지나지 않는 경우

✓ (a_x, a_y)와 (b_x, b_y)의 거리를 구하면 됩니다.



2. 오아시스를 지나는 경우

- ✓ (a_x, a_y) 에서 오아시스에 도달하기 위한 최단거리와,
- ✓ (b_x, b_y) 에서 오아시스에 도달하기 위한 최단거리를 더하면 됩니다.

- ✓ (a_x, a_y) 에 대한 거리는, (a_x, a_y) 와 (p_x, p_y) 의 거리에 r 을 뺀 값을 구하면 됩니다.
- ✓ 오아시스 내부에 점이 존재하는 경우에는 해당 값이 음수가 되니 0 으로 계산합시다.



D. 지정좌석제

`implementation` 출제진 의도 – **Easy**

- 제출 141번, 정답 27명 (정답률 19.858%)
- 처음 푼 사람: **혼돈을초월한우리의김태윤을찬양하라**, 30분
- 출제자: 이재찬 `dlwocks1209`

D. 지정좌석제



- 모든 자리를 기준으로 주변 $W \times W$ 범위를 탐색하면 시간복잡도는 O(RCW^2)으로 시간초과가 됩니다.
- 자리 기준으로 주변 범위를 탐색하는 것이 아닌, 친구들의 위치를 기준으로 주변 범위를 탐색하여 그 자리의 만족도를 1씩 올려야 합니다.
- 이 방법은 시간복잡도가 O($NW^2 + RC$)이며, 시간 내에 통과가 가능합니다.



E. Magical Trees

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

- 제출 38번, 정답 19명 (정답률 50%)
- 처음 푼 사람: **sugar**, 32분
- 출제자: 채성우 `lem0nad3`

E. Magical Trees



- N 이 짝수면 해 구성은 불가능합니다.
- $N = 2k$ (k는 양의 정수)면 간선 개수의 총합은 $3 \times (2k - 1) = 6k - 3$이며, 이는 홀수입니다.
- 따라서 두 번 등장하지 않는 간선은 무조건 존재합니다.

E. Magical Trees



- N 이 홀수면 어떨까요?
- $N = 2k - 1$ (k 는 양의 정수) 면 간선 개수의 총합은 $3 \times (2k - 2) = 6k - 6$ 이며, 이는 짝수입니다.
- 모든 간선이 두 번 등장하도록 간선을 정할 수 있습니다.

E. Magical Trees



- N 이 홀수인 경우, 가능한 구성 중 하나는 다음과 같습니다.

- $1 \leq x \leq N-2$ 인 정수 x 에 대해, 각 트리에 다음 간선을 추가합니다.
- 첫 번째 트리: $(x, x+1), (x, x+2)$
- 두 번째 트리: $(x, x+1), (x+1, x+2)$
- 세 번째 트리: $(x, x+2), (x+1, x+2)$

- 위 구성 방법은 모든 간선이 두 번만 등장하며, 트리임을 보장합니다.



F. 취향 변화

`math` 출제진 의도 – **Medium**

- 제출 34번, 정답 8명 (정답률 23.529%)
- 처음 푼 사람: **이람두꼬는나**, 80분
- 출제자: 강효규 `djs100201`

F. 취향 변화



- 취향이 변화하는 쿼리가 없다고 생각 해 봅시다.
- **좋아하는 물건의 개수 - 싫어하는 물건의 개수**는 항상 두 사람의 행복도의 합과 같습니다.
- 이 값을 S라고 해 봅시다.
- 즉 $dong_gas$의 행복도의 합과 $shandy5833$의 행복도의 합은 항상 S로 일정합니다.
- 합이 일정한 두 정수의 곱의 최댓값은 어떻게 될까요?
- 두 사람의 행복도가 가까울수록 커지게 됩니다!

F. 취향 변화



- 우리는 $dong_gas$ 의 행복도를 0 부터 S 까지 모든 정수 중 원하는 것으로 바꿀 수 있습니다.
- 왜 항상 그렇게 만들 수 있을까요?
- 잘 생각해보면 $dong_gas$ 의 행복도는 1 씩 변화합니다.
- $dong_gas$ 의 행복도는 0 으로 시작하여 모든 물건을 가져간다면 S 가 됩니다.
- 즉 행복도를 이 사이에 있는 모든 정수 들로 바꿀 수 있습니다.

F. 취향 변화



- 행복도가 $S/2$가 된다면 최적입니다.
- 그런 상황을 항상 만들 수 있음을 보였습니다.
- 따라서 각 쿼리의 정답은 S에만 관련이 있습니다.
- 전체 합을 S를 관리하면 문제를 해결 할 수 있습니다.
- 물건의 종류별 개수와 종류별 취향을 배열로 관리합시다.
- 쿼리마다 $O(1)$에 S를 정확히 추적할 수 있습니다.



G. LCS Making

`string`, `ad_hoc` 출제진 의도 – **Medium**

- 제출 26번, 정답 6명 (정답률 23.077%)
- 처음 푼 사람: **조다니엘카투사축하**, 74분
- 출제자: 채성우 `lem0nad3`

G. LCS Making



- $|LCS(s,t)|$가 최솟값을 가지도록 t를 구성해봅시다.
- 일반성을 잃지 않고, s에서 가장 적게 등장하는 알파벳을 'a'라고 합시다.
- t를 다음처럼 구성하면 $|LCS(s,t)|$는 최소입니다.
- $t = \underbrace{\text{aa...a}}_{N}$

G. LCS Making



- ✓ s에서 'a'가 등장하는 횟수를 cnt라고 합시다.
- ✓ $cnt \leq K \leq N$이라면 답은 1입니다. 아니라면 0입니다.

- ✓ $t = \underbrace{\text{aa...a}}_{N}$로 둔 뒤,
- ✓ $s[i] \neq$'a'라면 $t[i] \leftarrow s[i]$를 하는 것으로 $|LCS(s,t)|$를 1 늘릴 수 있습니다.



H. 점진적인 수열

dp 출제진 의도 – **Medium**

- 제출 13번, 정답 1명 (정답률 7.692%)
- 처음 푼 사람: **말년일병**, 159분
- 출제자: 강효규 `djs100201`

H. 점진적인 수열



- 수열 S가 서로 다른 원소들로 이루어져 있음을 이용합시다.
- 부분 수열을 이어 붙여 점진적인 수열을 만드는 방법을 생각해봅시다.
- 현재까지 만들어진 부분 수열에서 뒤의 두 개의 원소를 알고 있다면 다음에 올 수 있는 원소의 후보를 알 수 있습니다.
- 마지막 두 원소를 a와 b라고 해 봅시다.
- 이 다음 원소는 $2b-a-1, 2b-a, b-a+1$ 중 하나입니다.

H. 점진적인 수열



- 수열을 이어 붙여 만든다고 했을 때, 마지막 두 원소에 대한 2차원 dp를 생각해 봅시다.
- $dp[i][j]$를 S_i, S_j로 점진적인 수열을 시작했을 때 정답이라고 해 봅시다.
- $dp[i][j]$에서 최대 3가지로 전이가 됩니다.
- 전이시킬 때 가능한 경우의 수를 잘 세주면 문제가 풀립니다.
- $O(n^2)$ 혹은 $O(n^2 logn)$에 문제를 해결할 수 있습니다.



I. 광부가 될 수 있다면

`dp`, `prefix_sum` 출제진 의도 – **Medium**

- 제출 9번, 정답 0명 (정답률 0%)
- 처음 푼 사람: -, -분
- 출제자: 이재찬 `dlwocks1209`



- 이 문제는 DP, 누적합을 응용하여 해결할 수 있습니다.
- DP[i][j]를 현재 위치가 (i, j)일때 번 돈의 최댓값이라 하겠습니다.

- DP의 점화식을 구하기 위해서는 다음 정보들이 필요합니다.
- DPL[i][j]는 현재 위치가 (i, j)이고, 현재 층에서 왼쪽만 방문했을 때 번 돈의 최댓값입니다.
- DPR[i][j]는 현재 위치가 (i, j)이고, 현재 층에서 오른쪽만 방문했을 때 번 돈의 최댓값입니다.



- L[j]는 현재 층에서 j열부터 시작하여 왼쪽 방향으로의 최대 suffix sum입니다.
- R[j]는 현재 층에서 j열부터 시작하여 오른쪽 방향으로의 최대 prefix sum입니다.

- DPL과 DPR의 점화식은 다음과 같습니다.
- DPL[i][j] = max(DPL[i][j-1] + arr[i][j], DP[i-1][j] + L[j])
- DPR[i][j] = max(DPR[i][j+1] + arr[i][j], DP[i-1][j] + R[j])



- 마지막으로 DP의 점화식은 다음과 같습니다.
- DP[i][j] = max(DPL[i][j] + R[j], DPR[i][j] + L[i][j]) - arr[i][j]

- DP 마지막 행에서의 최댓값이 정답이 됩니다.



J. 거의 같은 문자열

String 출제진 의도 – **Hard**

- 제출 5번, 정답 0명 (정답률 0%)
- 처음 푼 사람: -, -분
- 출제자: 강효규 djs100201



- Q_i 가 단 하나일 때는 문제를 어떻게 풀까요?
- 이는 trie, 해싱, map 등의 다양한 방법으로 문제를 해결할 수 있습니다. 핵심은 Q_i 의 문자열 합에 대한 제한이 있다는 점입니다.
- Q_i 의 서로 다른 문자열 길이는 $O(\sqrt{M})$ 개 입니다.

J. 거의 같은 문자열



- 각 문자열의 길이를 고정하고 순회합시다.
- 이렇게 한다면, rolling hash와 같이 고정된 길이의 모든 연속부분문자열을 관리할 수 있습니다.
- 각 문자열 길이를 고정하고 앞에서 부터 각 문자에 대한 개수를 map이나 다른 자료구조로 관리하면 $O((N\sqrt{M}+MlogN)log26)$ 정도에 문제를 해결할 수 있습니다.
- 여기서 조심할 점은, $O(NlogN\sqrt{M})$ 의 시간복잡도의 구현은 시간초과를 받게 된다는 점입니다.



K. Ultimate A + B

`math`, `case_work`, `binary_search` 출제진 의도 – **Challenging**

- 제출 4번, 정답 0명 (정답률 0%)
- 처음 푼 사람: -, -분
- 출제자: 이재찬 `dlwocks1209`

K. Ultimate A + B



- N 개의 연산에 대해, 연산의 실제 값을 P 라고 하면 $K_i - E \le P \le K_i + E$ 입니다.
- 이 부등식들을 연립하면 결과적으로 P 의 범위, 즉 $A+B, A-B, A \times B$ 의 범위를 구할 수 있습니다.
- 단, A 와 B 가 양의 정수이므로 $A+B$ 는 2 이상, $A \times B$ 는 1 이상입니다.
- 편의상 각 연산의 범위를 $c_0 \le A+B \le c_1$, $c_2 \le A-B \le c_3$, $c_4 \le A \times B \le c_5$ 라 두겠습니다.
- 여기서 각 연산의 존재 여부에 따라 case를 나눠야 합니다.

K. Ultimate A + B



1. A+B만 있는 경우

- ✓ A의 값을 증가시키면 각 A에 대해 B의 범위를 구하고, 이를 모두 더해 순서쌍의 개수를 구할 수 있습니다.
- ✓ 각 A에 대한 B의 개수를 반복문으로 직접 더할 경우 시간초과가 발생하므로 수열의 합을 이용해 구해야 합니다.

- ✓ A가 1일 때 B의 범위는 $[c_0 - 1, c_1 - 1]$으로 $c_1 - c_0 + 1$ 개입니다.
- ✓ A가 $c_1 - 1$일 때 B의 범위는 $[1, 1]$으로 1개입니다.
- ✓ 정리하면 수식은 $(c_1 - c_0 + 1) \times (c_0 + c_1 - 2) \div 2$ 가 됩니다.



2. A-B만 있는 경우

- ✓ 이 case는 항상 답이 무한대입니다.
- ✓ A와 B의 값에 상한이 없으므로 A-B 범위를 만족하는 A와 B의 개수는 무한대가 됩니다.

K. Ultimate A + B



3. A+B, A-B만 있는 경우 (A×B가 없는 경우)

- ✓ 이 case도 1번 case와 비슷하게 A를 증가시키며 A에 대한 B의 범위를 모두 더하면 순서쌍의 개수를 구할 수 있습니다.
- ✓ 다만 직접 더할 경우 시간초과가 발생하며, B의 범위는 항상 선형적이지 않으므로 추가적인 작업이 필요합니다.

- ✓ A+B와 A-B의 범위에 대해 그래프를 그리면 직선이 서로 교차하는 부분을 기준으로 잘라 수열의 합을 구할 수 있습니다.
- ✓ B = c_0 - A, B = c_1 - A, B = A - c_2, B = A - c_3, B = 1이 교차하는 점들이 그 기준입니다.
- ✓ 이 점들의 시작과 끝, 길이를 이용하여 수열의 합을 구하면 결과적으로 (A, B)의 개수를 구할 수 있습니다.



4. A×B가 있는 경우

- ✓ 이 case도 이전과 비슷하게 A를 증가시키며 A에 대한 B의 범위를 모두 더하면 순서쌍의 개수를 구할 수 있습니다.
- ✓ 이 case 역시 직접 더할 경우 시간초과가 발생합니다.
- ✓ 대신, B의 범위가 바뀌는 지점들을 이분탐색을 이용하여 찾고, 범위와 길이를 이용하여 개수를 전부 더할 수 있습니다.



4. A×B가 있는 경우

- ✓ 이 방법이 시간초과가 발생하지 않는 이유는 다음과 같습니다.
- ✓ A가 $[1, \sqrt{c_5}]$일 때와 $[\sqrt{c_5}, c_5]$일 때로 나눠 확인해봅시다.
- ✓ 범위 $[1, \sqrt{c_5}]$에서는 $\sqrt{c_5}$ 만큼의 연산만 하면 됩니다.
- ✓ A가 $\sqrt{c_5}$일 때 B의 범위가 (???, $\sqrt{c_5}$)가 되는데, B의 범위는 이후 점점 작아지므로 A가 $[\sqrt{c_5}, c_5]$일 때 B의 범위가 바뀌는 지점은 $\sqrt{c_5}$를 넘지 않습니다.

- ✓ 결과적으로 O($\sqrt{N}\,logN$)의 시간복잡도에 해결이 가능합니다.