2025 HCPC

by 2025 ALOHA

Official Solutions

출제

✓ 김태우	mathtw1030	한양대학교 데이터사이언스전공 / ALOHA
✓ 노준혁	gggkik	한양대학교 컴퓨터소프트웨어학부 / ALOHA
✓ 박상혁	sangchoo1201	한양대학교 컴퓨터소프트웨어학부 / ALOHA
✓ 이상기	0xchaser	한양대학교 컴퓨터소프트웨어학부 / ALOHA
✓ 정성준	jsj0412	한양대학교 컴퓨터소프트웨어학부 / ALOHA

검수

- 79brue
- andrewmjk1
- azberjibiou
- hibye1217
- jthis
- leo020630
- lkh3happy
- mhy908
- myyh1234
- pizzaroot
- sadtreap
- terrasphere
- tlsdydaud1

Sponsors



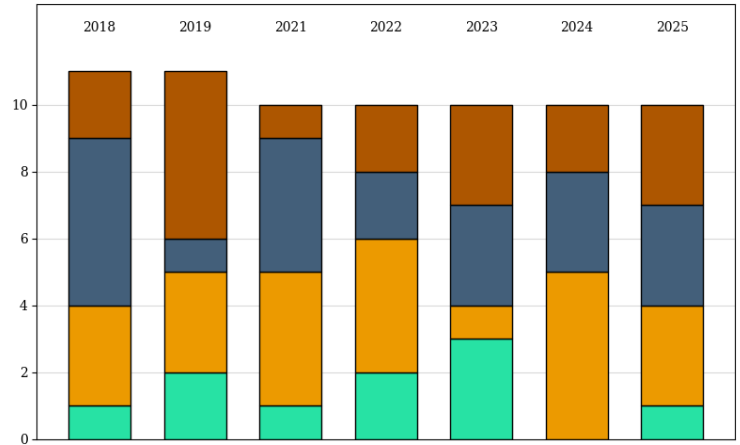
Sponsors



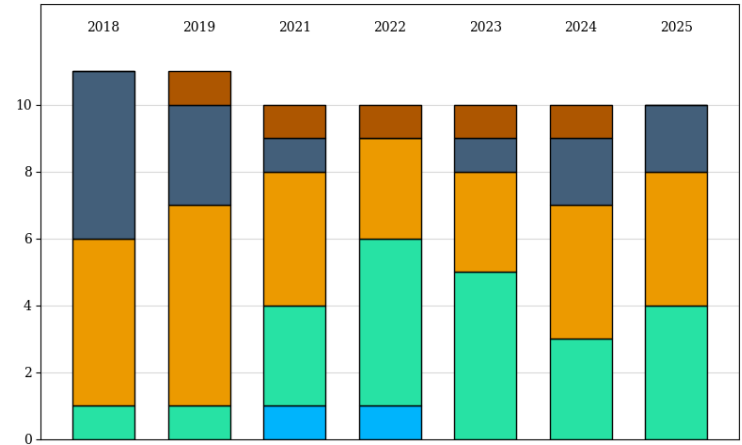
Sponsors



Difficulty Distribution - Beginner



Difficulty Distribution - Advanced



Beginner Division

문제		의도한 난이도	출제자
A	강의평	**Easy**	정성준
B	이웃 마을	**Hard**	정성준
C	꼬치	**Hard**	정성준
D	나이트와 킹	**Medium**	정성준
E	문자열 일치도	**Hard**	정성준
F	삼각형 그리기	**Medium**	정성준
G	저녁 메뉴 월드컵	**Easy**	김태우
H	이중 반복문	**Easy**	정성준
I	인버전의 기댓값	**Medium**	노준혁
J	특별한 수열	**Hard**	정성준

Advanced Division

문제		의도한 난이도	출제자
A	계단 오르고 내려오기	**Hard**	김태우
B	꼬치	**Medium**	정성준
C	문자열 일치도	**Medium**	정성준
D	삼각형 그리기	**Easy**	정성준
E	올바른 괄호 문자열 찾기	**Hard**	노준혁
F	인버전의 기댓값	**Easy**	노준혁
G	특별한 수열	**Medium**	정성준
H	해 구성 문제	**Challenging**	이상기
I	MEX들의 MEX	**Challenging**	노준혁
J	인터랙티브 XOR 게임	**Medium**	노준혁

BA. 강의평

`math, implementation`
출제진 의도 – **Easy**

- 처음 푼 팀: **코코코딩비기너**, 6분
- 출제자: 정성준

- 수를 하나씩 추가해보면서 평균이 X 를 넘었는지 검사해보면 됩니다.
- 이외에도 식을 정리해서 풀거나, 이진 탐색을 사용하는 등 다양한 풀이가 있습니다.

BB. 이웃 마을

`graphs`
출제진 의도 – **Hard**

- 처음 푼 팀: **joonyury**, 35분
- 출제자: 정성준

BB. 이웃 마을

- 주어진 쿼리를 단순히 수행할 경우 시간 초과가 발생합니다. 왜일까요?
- 1번 마을을 중심으로 연결된 성게 모양 그래프를 생각해봅시다.
- 2번 쿼리가 1번 마을로 들어올 때마다, $O(N)$ 의 연산을 수행해 검사해야 합니다. 따라서 총 $O(QN)$ 의 시간 복잡도를 가집니다.
- 간단한 트릭으로 시간 초과를 해결할 수 있습니다.
- 마을에 지하철역을 건설할 때, 그 마을의 모든 이웃 마을의 답에 1을 더합니다.
- 한 마을에 두 번 이상 건설되는 것을 무시하면, 시간 복잡도는 $O(M)$ 이 됩니다.
- 그래프의 모든 정점의 차수의 합은 간선 개수의 두 배와 같기 때문입니다.

BC, AB. 꼬치

`prefix_sum`, `two_pointer`
출제진 의도 – **Hard**

- 처음 푼 팀: **날긴하는비둘기**, 36분
- 출제자: 정성준

BC, AB. 꼬치

- 앞에서 a 개를 빼고 뒤에서 b 개를 빼서 정렬하려면, 다음 조건을 만족해야 합니다.
  - 앞 a 개의 수의 최댓값은 a 이다.
  - 뒤 b 개의 수의 최솟값은 $N - b + 1$ 이다.
  - 나머지 수들은 $A_i = i$ 이다.
- 앞에서부터 누적 최댓값, 뒤에서부터 누적 최솟값 배열을 계산합니다.
- 이후 두 포인터를 사용합니다.
- 왼쪽 포인터 i 에 대해, $A_j = j$ 인 동안 오른쪽 포인터 j 를 늘려 나갑니다.
- 이후 위의 1번째 조건과 2번째 조건을 검사한 뒤, 답을 업데이트하면 됩니다.

BD. 나이트와 킹

`ad_hoc`, `game_theory`, `parity`
출제진 의도 – **Medium**

- 처음 푼 팀: **풋붕쿤사실은우린내전우승해봤어**, 43분
- 출제자: 정성준

- 실제 체스판처럼 격자에 흑/백으로 색을 칠해 봅시다.
- 나이트는 항상 다른 색깔의 격자로 이동하는 것을 알 수 있습니다.
- 따라서, 킹은 항상 도착 지점과 같은 색깔의 칸으로 이동하면, 도착 지점에 말이 절대 도착하지 못하게 할 수 있습니다.
- 나이트가 선공이므로, 나이트가 이기는 유일한 경우는 시작하자마자 도착점에 도달 가능한 경우입니다.

BE, AC. 문자열 일치도

`binary_search`, `prefix_sum`
출제진 의도 – **Hard**

- 처음 푼 팀: **hibye1217**, 10분
- 출제자: 정성준

- 단순 $O(NM)$ 풀이는 시간 초과를 받습니다.
- 두 번째 문자열 T의 각 문자가 몇 번 일치될까요?
- T의 i번째 문자가 x라고 합시다. 이 문자가 일치되는 횟수는 곧 S의 i번째부터 $N-M+i$번째 문자 중 x가 등장하는 횟수와 같습니다.
- 이를 빠르게 구하기 위해 이분 탐색이나 누적 합을 사용할 수 있습니다.
- 이분 탐색을 사용할 경우 $O(N \log N)$, 누적 합을 사용할 경우 $O(26N)$ 입니다.

BF, AD. 삼각형 그리기

`math, combinatorics`
출제진 의도 – **Medium**

- 처음 푼 팀: **풋붕쿤사실은우린내전우승해봤어**, 29분
- 출제자: 정성준

BF, AD. 삼각형 그리기

- 겹치는 것을 고려하지 않으면, 삼각형은 ${}_NC_3$ 개입니다.
- 각 삼각형은 회전된 형태가 N 번 등장하므로, 답은 ${}_NC_3/N$ 입니다.
- 단, 정삼각형은 예외입니다. 정삼각형은 회전된 형태가 $N/3$ 번밖에 등장하지 않습니다.
- N 이 3의 배수라면, 답은 $\frac{{}_NC_3 - N/3}{N} + 1$ 입니다.
- 답이 int 범위를 넘을 수 있음에 주의합시다.

BG. 저녁 메뉴 월드컵

`math`

출제진 의도 – **Easy**

- 처음 푼 팀: **풋붕쿤사실은우린내전우승해봤어**, 5분
- 출제자: 김태우

- 각 테스트케이스에 대해 N 이 1 이 될 때까지 2 로 나누어 줍니다.
- 이때, N 이 홀수일 때 횟수를 세어 주면 됩니다.
- 시간 복잡도는 $O(TlogN)$ 입니다.

BH. 이중 반복문

`implementation`
출제진 의도 – **Easy**

- 처음 푼 팀: **반복문이멈추지않는탓일까ㅜ.ㅜ**, 5분
- 출제자: 정성준

BH. 이중 반복문

- 가장 쉬운 문제입니다.
- $N \times N$이 1억을 초과하는지 검사하거나, N이 1만을 초과하는지 검사하면 됩니다.
- $N \times N$을 계산할 경우, int 등의 자료형은 오버플로우가 발생할 수 있습니다.

BI, AF. 인버전의 기댓값

`math, ad_hoc`
출제진 의도 – **Medium**

- 처음 푼 팀: **NGM**, 8분
- 출제자: 노준혁

- $1 \leq i < j \leq n$ 인 임의의 (i, j) 쌍에 대하여 인버전이 될 가능성은 0.5 입니다.
- 가능한 (i, j) 쌍은 $n(n-1)/2$ 개이므로 인버전 개수의 기댓값은 $n(n-1)/4$ 입니다.
- DP를 이용한 $O(N^4)$ 별해가 존재합니다.
- $DP[i][j]$ 를 길이 i 의 순열에서 인버전의 개수가 j 개인 순열의 수라고 정의하여 풀면 됩니다.

BJ, AG. 특별한 수열

`math`, `prefix_sum`, `set`, `data_structure`
출제진 의도 – **Hard**

- 처음 푼 팀: **NGM**, 12분
- 출제자: 정성준

BJ, AG. 특별한 수열

- set을 사용하는 풀이와 누적 gcd를 사용하는 풀이로 총 두 가지 풀이가 있습니다.
- set을 사용하는 풀이입니다.
- 현재까지 추가된 수에 대한 multiset A와, 인접한 수들의 차이를 모두 담는 multiset B를 관리합시다.
- multiset에서 어떤 수보다 크면서 가장 작은 수, 어떤 수보다 작으면서 가장 큰 수를 $O(\log N)$에 구할 수 있습니다. set의 내장 lower_bound 함수를 사용하면 됩니다.
- 추가하려는 수의 앞/뒤 수의 차이를 A를 통해 구한 뒤, 이 차이를 B에서 제거합니다.
- 이후 새로 생기는 두 개의 차이를 B에 넣습니다.
- 특별한 수열인지 판별하는 것은 곧 B의 최솟값과 최댓값이 같은지 여부와 같습니다.

- 누적 gcd를 사용하는 풀이입니다.
- 수열이 특별한 수열이라는 것은 다음 조건을 만족하는 것과 동치입니다. 증명은 생략합니다.
  - (최댓값 - 최솟값) = (인접한 수들의 차의 gcd) $\times$ (수열 길이 - 1)을 만족하고, 모든 수가 다름
  - 또는, 모든 수가 같음
- 앞에서부터 누적 최솟값/최댓값, 누적 차 gcd, 수가 모두 다른지 여부를 관리하면 됩니다.

AA. 계단 오르고 내려오기

dp
출제진 의도 – **Hard**

- 처음 푼 팀: **N/A**, N/A분
- 출제자: 김태우

AA. 계단 오르고 내려오기

- 문제에서 구하는 값은 다음 경우의 수의 합입니다.
  - 중간에 K 개의 연속한 수가 없는 증가수열
  - 끝이 N 이거나, 끝이 $N-K+1$ 이고 수열에 $N-K$ 가 없음
- $DP[i][j]$ 를 $i-j+1, i-j+2, \cdots, i$ 를 포함하고 중간에 K 개의 연속한 수가 없는 증가수열의 수라고 합시다.
- 그리고 끝이 i 인 수열의 수 $DP[i][1]+DP[i][2]+...+DP[i][K-1]$ 를 $S[i]$ 라고 합시다.

- $DP[][], S[]$ 의 정의에 의해 다음이 성립합니다.
  - $DP[i][1] = S[i-2] + S[i-3] + ... + S[1] + 1 (i \leq K)$
  - $DP[i][1] = S[i-2] + S[i-3] + ... + S[i-K] (i > K)$
  - $DP[i][j] = DP[i-1][j-1] (j \geq 2)$
- $DP[i][1]$ 를 $A[i]$ 라고 합시다. (단, $i \leq 0$ 일 때 $A[i] = 0$)

- 이때, 다음이 성립합니다.

$$\begin{aligned} S[i] &= DP[i][1] + DP[i][2] + ... + DP[i][K-1] \\ &= DP[i][1] + DP[i-1][1] + ... + DP[i-K+2][1] \\ &= A[i] + A[i-1] + ... + A[i-K+2] \end{aligned}$$

- $A[]$와 $S[]$를 누적 합으로 관리하면 $O(N)$에 해결 가능합니다.

AA. 계단 오르고 내려오기

- 검수진 79brue님의 별해입니다.
- 각 계단을 올라갈 때 밟으면 U, 내려갈 때 밟으면 D를 적는다고 합시다.
- 길이 $N-1$의 문자열 S를 만들어, S가 U 또는 D로만 이루어져 있고, 같은 문자가 K번 연속 등장하지 않는 방법의 수로 환원 가능합니다.
- DP + 누적합을 이용해 $O(N)$에 계산 가능합니다.

AE. 올바른 괄호 문자열 찾기

ad_hoc
출제진 의도 – **Hard**

- 처음 푼 팀: **N/A**, N/A분
- 출제자: 노준혁

AE. 올바른 괄호 문자열 찾기

- 괄호 문자열은 (를 $+1$ 로,) 를 -1 로 생각할 수 있습니다.
- 올바른 괄호 문자열은 prefix sum 이 항상 0 이상이고 처음부터 끝까지의 합이 0 입니다.
- 문자열 S 를 2 개 이어붙인 것이기에, T_i 와 T_{2N+i} 는 항상 같습니다.
- 추가로, T 에서 연속한 $2N$ 개를 모두 더하면 0 이 나옵니다.
- 따라서 S 의 prefix sum 이 가장 작은 곳에서 시작한다면, 길이 $2N$ 의 올바른 괄호 문자열을 얻을 수 있습니다.
- 첫 번째 실행에서 최솟값이 위치한 인덱스와 최솟값을 넘기면, 두 번째 실행에서 이어서 확인할 수 있습니다.

AG. 해 구성 문제

`ad_hoc`, `constructive`, `case_work`
출제진 의도 – **Challenging**

- 처음 푼 팀: **방인규**, 78분
- 출제자: 이상기

- $l > 0$인 l에 대하여 $(A_i, A_{i+1}, A_{i+2}, A_{i+3}) = (l, -l, -l-1, l+1)$로 배정하면 합이 항상 1이 됩니다.
- 또한, 배정된 수의 부호를 모두 바꾸면 합이 -1이 되게 만들 수 있기에 8개의 합이 0이 되게 배정이 가능합니다.
- $0 \backsim 7$개의 수는 처음에 적절히 처리하고 남은 $8n$개의 수들은 위의 방법을 이용하면 모든 조건을 만족하는 해를 구성할 수 있습니다.

AI. MEX들의 MEX

`dp`, `bitmask`
출제진 의도 – **Challenging**

- ✓ 처음 푼 팀: **ALPYEONG**, 105분
- ✓ 출제자: 노준혁

AI. MEX들의 MEX

- 항상 답이 22 이하임을 증명할 수 있습니다.
- 이를 토대로 비트필드를 이용한 DP를 할 수 있습니다.
- 각 부분 수열의 mex가 i가 되는 경우가 등장했는지를 i번째 비트에 기록하면 됩니다.

AJ. 인터랙티브 XOR 게임

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

- 처음 푼 팀: **ALPYEONG**, 113분
- 출제자: 노준혁

- 후공으로 하는 것이 최선입니다.
- 선공이 x를 선택할 때 $y = x \oplus 1023$인 y를 선택하면 $x \oplus y = 1023$가 되게 할 수 있습니다.
- 임의의 x에 대하여 y가 유일하게 존재하기에 선택 가능함이 보장됩니다.
- n이 짝수면 min, 홀수면 max를 처음에 선택하고 위의 방식을 반복하면 됩니다.