제5회 디미고 프로그래밍 챌린지 해설

Official Solution

by

대회 운영진 일동

2026년 10월 26일

문제	문제 이름	예상 난이도	출제자
A	여기 대체 어디에 0과 1이 있다는 거냐?	**Bronze III**	k_san
B	문자열과 완전한 괄호	**Silver III**	munjeokman
C	조기 입실	**Silver III**	Mazeppa
D	Suspected-Sequence	**Silver II**	munjeokman
E	텃밭가꾸기	**Silver II**	wanthor
F	two convex sequences (easy)	**Silver I**	slow09
G	bfs나 다익이나	**Silver I**	k_san
H	이상한 집합	**Gold V**	myLady
I	선행 관계	**Gold V**	abcdana
J	와이파이 배정하기	**Gold III**	Mazeppa

문제	문제 이름	예상 난이도	출제자
K	감 따기	**Platinum IV**	aruba02_
L	잔류신청	**Platinum III**	thimphicp
M	압도적인 쌍	**Platinum III**	k_san
N	여기 어디야	**Platinum III**	k_san
O	함수열의 개수	**Platinum II**	myLady
P	small-big-sequence	**Platinum II**	wanthor
Q	two convex sequences (normal)	**Platinum II**	slow09
R	좀비고 고인물이 되고싶어!!	**Diamond V**	rumi348
S	two convex sequences (hard)	**Diamond IV**	slow09
EXTRA	이 트리 뭐야	**Unrated**	k_san

A. 여기 대체 어디에 0과 1이 있다는 거냐?

`implementation string`
예상 난이도 – **Bronze III**

- 본 대회 – 제출 횟수 82회 · 정답자 수 32명 · 정답률 59.76%
- Open Contest – 제출 횟수 24회 · 정답자 수 19명 · 정답률 79.17%
- 최초 정답자(본 대회): 2421 (1분)
- 최초 정답자(Open Contest): do_manosaba (1분)
- 출제자: k_san

A. 여기 대체 어디에 0과 1이 있다는 거냐?

- 문자열을 왼쪽부터 확인하며, 현재 문자와 다음 문자가 '01'이면 '@'를 출력하고 두 칸 이동한다.
- 그렇지 않으면 현재 문자를 그대로 출력하고 한 칸 이동한다. 문자열의 끝까지 이 과정을 반복한다.
- 파이썬에서는 `S.replace('01', '@')`의 결과를 출력하면 된다.

B. 문자열과 완전한 괄호

`string` `ad_hoc`
예상 난이도 – **Silver III**

- 본 대회 – 제출 횟수 71회 · 정답자 수 21명 · 정답률 52.11%
- Open Contest – 제출 횟수 12회 · 정답자 수 8명 · 정답률 66.67%
- 최초 정답자(본 대회): 1418 (29분)
- 최초 정답자(Open Contest): `AdTerram` (6분)
- 출제자: `munjeokman`

B. 문자열과 완전한 괄호

- 같은 알파벳 두 개는 한 번의 연산으로 ( ) 로 바꿀 수 있다.
- 각 알파벳의 등장 순서대로 두 개씩 묶어 앞쪽을 (, 뒤쪽을) 로 바꾸면 완전한 괄호 문자열이 된다.
- 따라서 가능한 한 많은 쌍을 만들고, 남은 문자는 제거하면 된다.

  알파벳 c 의 등장 횟수를 cnt_c 라 하면 정답은

$$\boxed{N - \sum_{c='a'}^{'z'} \left\lfloor \frac{cnt_c}{2} \right\rfloor}$$

  시간 복잡도는 $O(N)$ 이다.

C. 조기 입실

ad_hoc
예상 난이도 – **Silver III**

- 본 대회 – 제출 횟수 160회 · 정답자 수 17명 · 정답률 10.63%
- Open Contest – 제출 횟수 15회 · 정답자 수 5명 · 정답률 33.33%
- 최초 정답자(본 대회): 2415 (18분)
- 최초 정답자(Open Contest): do_manosaba (9분)
- 출제자: Mazeppa

C. 조기 입실

- 목표 달성이 가능한 경우에는 연산 순서와 관계없이 필요한 연산 횟수가 같으므로, 목표 달성이 불가능한 경우만 판별하면 된다.
- 문자열에 1의 개수가 홀수인 `0`1…1`0` 꼴의 부분 문자열이 있거나, 1의 개수가 홀수인 `0`1…1 꼴의 접미사가 있으면 목표를 달성할 수 없다.
- 목표 달성이 가능한 경우, 필요한 연산 횟수는 다음과 같이 계산할 수 있다.
- 문자열에 등장하는 1의 개수를 K, 처음에 1들이 놓인 위치의 인덱스 합을 S라 하자. 1을 모두 왼쪽으로 모았을 때의 인덱스 합을 T라 하면, $T = 1 + 2 + \cdots + K$이다.
- 새치기를 한 번 할 때마다 1들의 인덱스 합이 2씩 줄어드므로, 필요한 연산 횟수는 $\frac{S - T}{2}$이다.

D. Suspected-Sequence

`math`
예상 난이도 – **Silver II**

- 본 대회 – 제출 횟수 38회 · 정답자 수 16명 · 정답률 44.74%
- Open Contest – 제출 횟수 6회 · 정답자 수 5명 · 정답률 83.33%
- 최초 정답자(본 대회): 2421 (23분)
- 최초 정답자(Open Contest): do_manosaba (5분)
- 출제자: `munjeokman`

D. Suspected-Sequence

- ✓ $\lfloor x \rfloor = n$ 이라 하면 $n \le x < n+1$ 이다.
- ✓ $\dfrac{x^2}{n}$ 이 양의 정수여야 하므로 가능한 x^2 의 값은 $n^2, n(n+1), n(n+2)$ 이다.
- ✓ 따라서 각 정수 n 마다 정확히 3개의 항이 존재한다.

k 번째 항에 대해

$$n = \left\lfloor \frac{k-1}{3} \right\rfloor + 1, \qquad r = (k-1) \bmod 3$$

으로 계산하면 정답은

$$\boxed{N^2 = n(n+r)}$$

시간 복잡도는 테스트 케이스당 $O(1)$ 이다.

E. 텃밭가꾸기

`sorting` `ad_hoc`
예상 난이도 – **Silver II**

- 본 대회 – 제출 횟수 70회 · 정답자 수 13명 · 정답률 18.57%
- Open Contest – 제출 횟수 5회 · 정답자 수 4명 · 정답률 80.00%
- 최초 정답자(본 대회): 2421 (33분)
- 최초 정답자(Open Contest): do_manosaba (15분)
- 출제자: `wanthor`

E. 텃밭가꾸기

- 모든 묘목의 높이가 다르므로 두 개의 묘목을 배열했을 때에는 반드시 하나의 묘목은 가려지게 된다.
- 이때 모든 묘목의 초기 생산량이 같기 때문에, 높이가 높은 나무가 최대한 다른 나무에게 가려지지 않도록 하는 것이 좋다.
- 따라서 높이가 높은 상위 $\lfloor N/2 \rfloor$ 개의 묘목을 먼저 배치한 후, 사이사이에 하위 $\lfloor N/2 \rfloor$ 개를 배치하여 지그재그 수열을 구성하면 가장 높이가 낮은 나무들만 가려져 생산량의 최댓값을 구할 수 있다.
- 주어진 나무를 정렬하여 $O(NlogN)$ 에 해결할 수 있다.

F. two convex sequences (easy)

`math` `ad_hoc`

예상 난이도 – **Silver I**

- 본 대회 – 제출 횟수 42회 · 정답자 수 5명 · 정답률 16.67%
- Open Contest – 제출 횟수 5회 · 정답자 수 3명 · 정답률 60.00%
- 최초 정답자(본 대회): 2421 (203분)
- 최초 정답자(Open Contest): do_manosaba (39분)
- 출제자: `slow09`

F. two convex sequences (easy)

- easy에서는 $M = 0$이므로 모든 i에 대해 $AL_i = BL_i, AR_i = BR_i$이고, 두 수열의 첫 항과 마지막 항도 같은 값으로 고정되어 있다.
- 양 끝점을 잇는 등차수열 C를 정의하자.

$$C_i = \frac{(N-i)AL_1 + (i-1)AL_N}{N-1}.$$

F. two convex sequences (easy)

- A는 아래로 볼록하므로 $A_i \le C_i$이고, B는 위로 볼록하므로 $C_i \le B_i$이다. 따라서 모든 i에 대해 $AL_i \le C_i \le AR_i$여야 한다.
- 반대로 이 조건을 만족하면 $A = B = C$로 선택할 수 있으므로, 조건은 필요충분하다.
- 분모 $N-1$을 곱해

$$(N-1)AL_i \le (N-i)AL_1 + (i-1)AL_N \le (N-1)AR_i$$

를 정수 연산으로 확인하면 실수 오차 없이 $\mathcal{O}(N)$에 해결할 수 있다.

G. bfs나 다익이나

`graphs` `ad_hoc` `constructive` `graph_traversal` `bfs`
예상 난이도 – **Silver I**

- 본 대회 – 제출 횟수 13회 · 정답자 수 4명 · 정답률 30.77%
- Open Contest – 제출 횟수 9회 · 정답자 수 3명 · 정답률 33.33%
- 최초 정답자(본 대회): 2421 (766분)
- 최초 정답자(Open Contest): `tyranno_zzang_ssem` (47분)
- 출제자: `k_san`

G. bfs나 다익이나

- BFS에서 새 정점을 발견하는 간선에 1 부터 $N-1$ 까지 차례대로 부여한다. 나머지 간선에는 남은 가중치를 부여한다.
- 트리 경로의 가중치 합을 $D(v)$ 라 하자.
- 비트리 간선 (u, v) 에서 u 가 먼저 발견되었다면, v 의 부모는 u 보다 먼저 처리된다. 따라서 $0 < D(v) - D(u) \leq N-1$ 이다.
- 비트리 간선의 가중치는 N 이상이므로 트리 경로보다 짧아질 수 없다. 즉, $d(v) = D(v)$ 이다.
- BFS는 $e(v)$ 가 작은 순서로 정점을 발견하므로 조건을 만족한다.

H. 이상한 집합

`set`

예상 난이도 – **Gold V**

- 본 대회 – 제출 횟수 35회 · 정답자 수 6명 · 정답률 17.14%
- Open Contest – 제출 횟수 1회 · 정답자 수 1명 · 정답률 100.00%
- 최초 정답자(본 대회): 2421 (587분)
- 최초 정답자(Open Contest): do_manosaba (27분)
- 출제자: myLady

H. 이상한 집합

- 문제에서 주어진 이상한 집합이 될 조건 $\lfloor \frac{n}{k} \rfloor \notin S \rightarrow n \notin S$ 를 생각하자
- 이 명제의 대우는 $n \in S \rightarrow \lfloor \frac{n}{k} \rfloor \in S$ 이다.
- X 의 임의의 원소 v 에 대해서, $v_1 = v, v_i = \lfloor \frac{v_{i-1}}{k} \rfloor$ 인 수열을 생각하자
- v 가 S 의 원소이기 위해서는 v_1 이 S 의 원소여야 하고,
- v_1 가 S 의 원소이기 위해서는 v_2 이 S 의 원소여야 하고...
- 이렇게 가다가 $v_i = 0$ 이면 멈추게 된다.

H. 이상한 집합

- 이제 v에 대해서 우리가 정의한 수열에 대해, 집합 $\{v_1, v_2, \ldots, 0\}$를 $p(v)$라고 정의하자
- v가 S의 원소여야 한다는 조건만 고려하면 $p(v) \subseteq S$만으로 충분하다
- 이제 이를 S의 모든 원소에 대해서 고려하면 조건을 만족시키는 최소 크기의 S는 $\bigcup_{x \in X} p(x)$ 임을 알 수 있다.
- 이를 계산하기 위해서 map을 관리하면서 맵의 key값에는 S의 원소들을, value에는 $x \in p(key)$인 X의 원소 x의 개수를 관리하면서 풀 수 있다.
- 시간복잡도는 $O(Q \log 10^{17})$이다.
- 여담으로, 이 문제는 $\lfloor \frac{v}{k} \rfloor$가 v의 부모가 되는 트리로 모델링하면 주어진 원소들을 모두 포함하는 최소 크기의 트라이를 구하는 문제로 볼 수 있다.

I. 선행 관계

`data_structure stack`
예상 난이도 – **Gold V**

- 본 대회 – 제출 횟수 42회 · 정답자 수 6명 · 정답률 14.29%
- Open Contest – 제출 횟수 1회 · 정답자 수 1명 · 정답률 100.00%
- 최초 정답자(본 대회): 1417 (408분)
- 최초 정답자(Open Contest): do_manosaba (45분)
- 출제자: abcdana

I. 선행 관계

- i가 j의 선행 프로젝트라는 것은, i의 오른쪽에서 처음으로 A_i보다 큰 값이 나오는 위치가 j라는 뜻입니다.
- 왼쪽부터 차례로 살펴보며, 아직 자신보다 큰 값을 만나지 못한 프로젝트의 번호를 스택에 저장합시다. 스택의 난이도는 아래에서 위로 갈수록 감소하도록 유지합니다.
- 현재 프로젝트가 i일 때, 스택의 맨 위에 있는 j가 $A_j < A_i$를 만족하는 동안 j를 꺼냅니다. j가 처음 만난 더 어려운 프로젝트가 i이므로, 이때 꺼내는 프로젝트들이 정확히 i의 선행 프로젝트입니다.

- 꺼낸 프로젝트들의 실행 단계 중 최댓값에 1을 더하면 t_i가 됩니다. 꺼낸 프로젝트가 없다면 $t_i = 1$입니다. 계산을 마친 뒤에는 i를 스택에 넣습니다.
- 선행 프로젝트는 항상 앞에 있으므로, 필요한 실행 단계는 이미 계산되어 있습니다. 따라서 답은 항상 존재하며, -1을 출력할 일은 없습니다.
- 각 프로젝트는 스택에 한 번 들어가고 많아야 한 번 꺼내집니다. 따라서 시간 복잡도와 공간 복잡도는 모두 $O(N)$입니다.

J. 와이파이 배정하기

`bruteforcing` `greedy`

예상 난이도 – **Gold III**

- 본 대회 – 제출 횟수 14회 · 정답자 수 6명 · 정답률 50.00%
- Open Contest – 제출 횟수 1회 · 정답자 수 1명 · 정답률 100.00%
- 최초 정답자(본 대회): 2421 (63분)
- 최초 정답자(Open Contest): do_manosaba (87분)
- 출제자: Mazeppa

J. 와이파이 배정하기

- 학생들을 와이파이에 배정할 때는 r_i 가 작은 학생들 부터 고려하는 것이 이득이다.
- 와이파이의 대역폭이 큰 순서 혹은 작은 순서대로 학생들을 넣는 방법을 생각해볼 수 있지만, 이러한 그리디적인 접근법에는 반례가 생긴다.
- 와이파이가 3개밖에 되지 않으므로, $3! = 6$ 개의 와이파이 순열 각각에 대해 r_i 가 작은 학생부터 차례로 배정해보면 된다.
- 학생들은 선형적으로 $O(N)$ 에 배정해도 되고, 배정 가능한 학생 수로 매개 변수 탐색을 이용해 $O(\log N)$ 에 최댓값을 찾을 수도 있다.

K. 감 따기

`data_structure` `sorting` `trees` `euler_tour_technique` `segtree` `merge_sort_tree` `binary_search`

예상 난이도 – **Platinum IV**

- 본 대회 – 제출 횟수 46회 · 정답자 수 3명 · 정답률 34.78%
- Open Contest – 제출 횟수 4회 · 정답자 수 1명 · 정답률 25.00%
- 최초 정답자(본 대회): 2415 (221분)
- 최초 정답자(Open Contest): do_manosaba (139분)
- 출제자: aruba02_

K. 감 따기



- 루트에서 거리 H 이내의 리프는 미리 세어 둔다. 이동한 뒤 새로 얻는 감은 이동한 노드의 서브트리 안에서, 루트로부터는 H 보다 멀지만 그 노드로부터는 H 이내인 리프이다.
- 답은 처음 딴 감에, 이동한 노드에서 새로 얻는 감의 수의 최댓값을 더한 것이다.
- 노드마다 서브트리를 훑을 수는 없으므로 오일러 경로 테크닉(ETT)을 사용한다. DFS 방문 순서로 번호를 매기면 각 서브트리는 배열의 연속 구간이 된다.
- 루트에서 거리가 H 보다 먼 리프의 칸에만 루트로부터의 거리를 기록하고, 나머지 칸은 아주 큰 값으로 둔다.

K. 감 따기



- 각 노드에서 새로 얻는 감의 수는 서브트리 구간 안에서 값이 해당 노드의 거리와 H 의 합 이하인 칸의 개수이다.
- 이 개수는 머지 소트 트리와 이분 탐색으로 구한다. 구간을 세그먼트 트리의 $O(\log N)$ 개 노드로 나누고, 각 노드에서 이분 탐색으로 개수를 센다.
- 쿼리 하나당 시간 복잡도는 $O(\log^2 N)$ 이므로, 모든 노드에 대한 전체 시간 복잡도는 $O(N \log^2 N)$ 이다.

L. 잔류신청

`data_structure` `segtree` `binary_search`

예상 난이도 – **Platinum III**

- 본 대회 – 제출 횟수 68회 · 정답자 수 3명 · 정답률 4.41%
- Open Contest – 제출 횟수 5회 · 정답자 수 2명 · 정답률 40.00%
- 최초 정답자(본 대회): 2421 (547분)
- 최초 정답자(Open Contest): do_manosaba (95분)
- 출제자: `thimphicp`

L. 잔류신청

- 길이가 K 이상이고 원소의 합이 S 이상인 연속 구간 중, 최댓값과 최솟값의 차이가 가장 작은 구간을 찾아야 한다.

잔류 신청

- 구간 $[l, r]$ 의 길이가 K 이상이고, 원소의 합이 S 이상이어야 한다.
- 이 조건을 만족하는 구간 중 $\max(A_l, \ldots, A_r) - \min(A_l, \ldots, A_r)$ 의 최솟값을 구해야 한다.
- 불안정도의 상한을 X 로 정하고, 조건을 만족하면서 불안정도가 X 이하인 구간이 존재하는지 판별하자.
- 어떤 X 에서 가능한 구간이 존재한다면, 그보다 큰 X 에서도 항상 가능하다.
- 따라서 정답에 대한 이분 탐색을 사용할 수 있다.

잔류 신청

- 오른쪽 끝점 r을 고정하고, 불안정도가 X 이하인 가장 긴 구간 $[l, r]$을 구하자.
- 구간의 최댓값과 최솟값을 각각 단조 덱으로 관리하면, 투 포인터를 이용해 모든 r에 대한 l을 $O(N)$에 구할 수 있다.
- 구간 $[l, r]$의 부분 구간은 불안정도가 X 이하이므로, 시작점 s는 $s \geq l$을 만족하면 된다.
- 또한 구간의 길이가 K 이상이어야 하므로 $s \leq r - K + 1$이다.
- 따라서 가능한 시작점의 범위는 $l \leq s \leq r - K + 1$이다.

잔류 신청

- 누적합 $P_i = \sum_{j=1}^{i} A_j$ 를 정의하자. 단, $P_0 = 0$ 이다.
- 구간 $[s, r]$ 의 합이 S 이상이려면 $P_r - P_{s-1} \geq S$, 즉 $P_{s-1} \leq P_r - S$ 여야 한다.
- 가능한 시작점의 범위가 $l \leq s \leq r - K + 1$ 이므로, 다음 조건을 확인하면 된다.

$$\min_{l-1 \leq j \leq r-K} P_j \leq P_r - S$$

- 누적합 배열에 대한 구간 최솟값 세그먼트 트리를 구성하면, 위 조건을 $O(\log N)$ 에 판별할 수 있다.
- 모든 r 에 대해 검사하므로, 한 번의 판정에 $O(N \log N)$ 이 필요하다.

잔류 신청

- 불안정도 X 의 범위는 $0 \leq X \leq \max(A) - \min(A)$ 이다.
- 이 범위에서 이분 탐색하여 조건을 만족하는 가장 작은 X 를 구한다.
- 최댓값과 최솟값의 차이에서도 조건을 만족하는 구간이 없다면 -1 을 출력한다.
- 각 판정에서 투 포인터와 단조 덱에 $O(N)$, 세그먼트 트리 질의에 $O(N \log N)$ 이 필요하다.
- 따라서 전체 시간 복잡도는 $O(N \log N \log V)$ 이다. 여기서 V 는 원소 값의 범위이다.

M. 압도적인 쌍

ad_hoc
예상 난이도 – **Platinum III**

- 본 대회 – 제출 횟수 8회 · 정답자 수 2명 · 정답률 25.00%
- Open Contest – 제출 횟수 12회 · 정답자 수 2명 · 정답률 16.67%
- 최초 정답자(본 대회): 2415 (99분)
- 최초 정답자(Open Contest): `rladbeka` (62분)
- 출제자: `k_san`

M. 압도적인 쌍

- 압도적인 쌍의 조건을 만족하는 쌍의 개수가 많지 않다.
- 모든 쌍을 일일이 세줘도 되지 않을까?
- 각 i에서 양쪽으로 탐색하며, A_i가 큰 끝점인 쌍만 처리한다. 값이 같으면 왼쪽 끝점에서만 처리한다.
- 모든 원소가 양수이므로, 내부 합 S가 $2A_i$를 넘으면 탐색을 중단한다.

M. 압도적인 쌍



- 끝점 하나를 고정하고 한쪽에서 그곳까지 탐색하는 시작점들을 가까운 순서로 보자.
- 더 먼 시작점의 내부에는 앞선 시작점들이 모두 포함된다. 따라서 그 값은 앞선 값들의 합의 절반 이상이어야 한다.
- 즉, 시작점이 추가될 때마다 값의 합이 최소 $3/2$ 배로 늘어난다.
- 원소의 최댓값을 X 라 하면 합이 $2X$ 를 넘은 뒤에는 다음 시작점이 나올 수 없다.
- 따라서 한 끝점에 도달하는 시작점은 $O(\log X)$ 개이고, 전체 후보 쌍은 $O(N \log X)$ 개이다.

N. 여기 어디야

`game_theory` `implementation` `simulation`
예상 난이도 – **Platinum III**

- 본 대회 – 제출 횟수 18회 · 정답자 수 2명 · 정답률 11.11%
- Open Contest – 제출 횟수 12회 · 정답자 수 2명 · 정답률 16.67%
- 최초 정답자(본 대회): 2421 (836분)
- 최초 정답자(Open Contest): `rladbeka` (253분)
- 출제자: `k_san`

N. 여기 어디야

- 색과 지금까지의 차수 기록이 같은 정점들은 서로 구별할 수 없다.
- 이 정점들을 그룹으로 묶으면 크기가 1인 그룹만 탈출할 수 있다.
- 처음에는 색과 차수로 그룹을 나누고 이후에는 각 그룹을 직전 턴에 잃은 이웃의 수로 다시 나누는 것을 반복한다.
- 모든 분할을 마친 뒤 크기가 1인 그룹을 동시에 탈출시킨다.
- 아무도 탈출하지 않으면 정보가 더 변하지 않으므로 탈출하지 못한 정점의 정답은 -1이다.

N. 여기 어디야

- 탈출한 정점의 이웃만 큐에 넣고 잃은 이웃 수를 센다.
- 기존 그룹별로 `map`을 이용해 잃은 이웃 수가 같은 정점들을 묶는다.
- 크기가 1이 된 그룹의 정점을 다음 턴의 탈출 대상으로 모은다.
- 모든 간선은 각각 상수 번만 확인하므로 전체 시간 복잡도는 $O((N+M)\log N)$이다.

O. 함수열의 개수

분류 – math dp combinatorics exponentation_by_squaring
예상 난이도 – **Platinum II**

- 본 대회 – 제출 횟수 2회 · 정답자 수 1명 · 정답률 50.00%
- Open Contest – 제출 횟수 2회 · 정답자 수 1명 · 정답률 50.00%
- 최초 정답자(본 대회): 2421 (740분)
- 최초 정답자(Open Contest): do_manosaba (210분)
- 출제자: myLady

O. 함수열의 개수

- 정답을 구하는 dp를 생각해보자
- $dp[i][j][k]$는 k번 함수를 합성했을 때 정의역의 원소를 i개만 고려했을 때 그들의 치역의 원소가 j개가 되는 경우의 수로 정의한다(이 때, 치역은 원소가 j개가 되는 집합은 모두 서로 같은 것으로 보아, 구분하지 않고 하나로 센다).
- 이렇게 하면 정답은 $N \times dp[N][1][K]$이다.

O. 함수열의 개수

- 이 때, 함수를 $a+b$번 합성 한 것은 함수를 a번 합성하고, b번 합성한 것을 합성한 것으로 볼 수 있다.
- 이를 바탕으로 전이를 생각해보면 $dp[i][j][a+b]$ 는, a번 합성했을 때 함수값의 개수가 k개가 된다고 생각하면
- 중간에 k개가 될 원소들을 결정하는 경우의 수를 곱해줘야 하므로
- $dp[i][j][a+b] = \sum_{k=1}^{N}(dp[i][k][a] \times dp[k][j][b] \times \binom{N}{k})$ 이다.

- 또한, $dp[i][j][1]$ 은 i 개의 원소를 j 개의 집합으로 나눠서 각 함수값에 할당하고, 남은 원소들을 아무렇게나 배치하는 경우의 수이므로 $j! \times S_2(i, j) \times N^{N-i}$ 이다.
- 분할정복을 이용해서 k 번째 dp배열을 $\lfloor \frac{k}{2} \rfloor$ 번째 dp배열을 구해서 합침으로써 구하면 $O(N^3 \log K)$ 에 답을 구할 수 있다.

P. small-big-sequence

분류 – `dp` `traceback`
예상 난이도 – **Platinum II**

- 본 대회 – 제출 횟수 52회 · 정답자 수 2명 · 정답률 3.85%
- Open Contest – 제출 횟수 1회 · 정답자 수 0명 · 정답률 0.00%
- 최초 정답자(본 대회): 2421 (194분)
- 최초 정답자(Open Contest): 없음
- 출제자: `wanthor`

- 가려지지 않고 주어지는 마지막 숫자를 X 라 하자.
- dp의 정의를 생각해보면 $dp[i][j]$ 는 현재 왼쪽부터 i 개의 숫자를 복원했고, 사용하지 않은 수 중 X 보다 작은 수가 j 개 남아있을 때의 **복원 가능 여부**(`true`/`false`)로 설정할 수 있다
- 입력으로 주어지는 수열은 항상 복원 가능하기 때문에 기저상태는 $dp[N-1][0] =$ true 이며, 상태 전이는 인덱스가 작아지는 방향($N-1 \to 0$), 수열 복원은 인덱스가 커지는 방향 ($0 \to N-1$)으로 진행된다. $O(N^2)$

- 아직 사용하지 않은 r 개의 수를 정렬할 때, $lim = \lfloor r/2 \rfloor$ 라 하면 상대적인 rank가 $\leq lim$ 일 때 S(small), $> lim$ 일 때 B(big)이다.

$$\underbrace{[\quad \text{S} \quad]}_{lim} \quad \underbrace{[\quad \text{B} \quad]}_{r-lim}$$

- dp 테이블은 X 보다 큰 수 중 가장 큰 수, X 보다 작은 수 중 가장 작은 수만을 고려한다.

- **S인 경우**: S인 범위에서 X보다 큰 수를 고르는 경우와 작은 수를 고르는 경우를 모두 생각해준다.

$$dp[i][j] = dp[i+1][j-1]\ (j > 0)\ \vee\ dp[i+1][j]\ (j < lim)$$

- **B인 경우**: B인 범위에서도 동일하게 전이시킨다.

$$dp[i][j] = dp[i+1][j-1]\ (j > lim+1)\ \vee\ dp[i+1][j]\ (j < r-1)$$

- **복원**: 사전순으로 가장 작은 경우부터 $dp[i][j-1]$ 또는 $dp[i][j]$ 가 참인 항목을 찾아 앞에서부터복원한다. ($O(N^2)$)

Q. two convex sequences (normal)

`geometry` `binary_search` `parametric_search` `convex_hull`

예상 난이도 – **Platinum II**

- 본 대회 – 제출 횟수 14회 · 정답자 수 1명 · 정답률 7.14%
- Open Contest – 제출 횟수 0회 · 정답자 수 0명 · 정답률 —
- 최초 정답자(본 대회): 2421 (904분)
- 최초 정답자(Open Contest): 없음
- 출제자: `slow09`

Q. two convex sequences (normal)

- normal에서도 $M = 0$이지만 양 끝점은 고정되어 있지 않다. 먼저 공통 첫 항 $A_1 = B_1 = x$를 고정했을 때를 생각하자.
- 일반적으로 $L_i \le X_i \le R_i$인 아래로 볼록한 수열 X가 존재하는지 판단한다. 첫 항은 $L_1 = R_1 = x$로 고정한다.
- 점 (i, R_i)들의 아래쪽 볼록 껍질을 만들고, 인접한 꼭짓점을 선분으로 연결하자. 각 정수 위치의 선분 값을 H_i라 하자.
- H는 아래로 볼록하고 모든 위치에서 $H_i \le R_i$를 만족한다. 또한 조건을 만족하는 아래로 볼록한 수열 중 각 위치에서 가장 큰 수열이다.

Q. two convex sequences (normal)

- 껍질의 인접한 꼭짓점 인덱스를 $a < b$라 하자. 가능한 임의의 수열 X에 대해 $a \le j \le b$에서

$$X_j \le \frac{b-j}{b-a}X_a + \frac{j-a}{b-a}X_b \le \frac{b-j}{b-a}R_a + \frac{j-a}{b-a}R_b = H_j$$

- 따라서 어떤 위치에서 $H_i < L_i$이면 불가능하다. 모든 위치에서 $H_i \ge L_i$이면 $X = H$로 선택하면 된다.

Q. two convex sequences (normal)

- 아래쪽 볼록 껍질은 인덱스 순서대로 점을 추가하고, 선분 기울기가 증가하도록 스택을 관리해 구성한다.
- 각 선분 내부의 위치에서 다음 부등식을 검사하면 실수 연산이 필요 없다.

$$L_j(b-a) \le R_a(b-j) + R_b(j-a).$$

- 각 점은 스택에 한 번 들어가고 최대 한 번 제거된다. 선분 검사도 전체 $O(N)$ 이므로 고정한 첫 항에 대한 판정은 $O(N)$ 이다.

Q. two convex sequences (normal)

- ✓ B는 부호를 뒤집어 처리한다. $-B$는 아래로 볼록하고, 범위는 $-BR_i \le -B_i \le -BL_i$이다.
- ✓ 공통 첫 항의 후보 범위는 다음과 같다. $\ell > r$이면 불가능하다.

$$\ell = \max(AL_1, BL_1), \qquad r = \min(AR_1, BR_1).$$

- ✓ 범위 안에서 A의 첫 항을 높이면 조건을 만족하기 쉽고, B의 첫 항을 낮춰도 조건을 만족하기 쉬워진다. 이 단조성으로 가능한 최소 A_1인 α와 최대 B_1인 β를 각각 이분 탐색으로 구한다.
- ✓ 한쪽이라도 가능한 첫 항이 없거나 $\alpha > \beta$이면 불가능하다. 그렇지 않으면 공통 첫 항을 선택할 수 있다.
- ✓ 후보 범위의 크기를 V라 하면 시간 복잡도는 $\mathcal{O}(N \log V)$, 공간 복잡도는 $O(N)$이다.

R. 좀비고 고인물이 되고싶어!!

분류 – `geometry` `convex_hull` `tree` `binary_search`
예상 난이도 – **Diamond V**

- 본 대회 – 제출 횟수 30회 · 정답자 수 1명 · 정답률 6.67%
- Open Contest – 제출 횟수 0회 · 정답자 수 0명 · 정답률 —
- 최초 정답자(본 대회): 2421 (1,528분)
- 최초 정답자(Open Contest): 없음
- 출제자: `rumi348`

- 각 정점 i를 좌표평면의 점

$$P_i = (A_i, \text{depth}[i])$$

로 나타내자. 한 경로 위의 정점들을 모두 같은 평면에 놓고 생각한다.

- 경로에서 조상 i에서 자손 j로 갈 때 두 점 사이의 비용은 두 점이 만드는 사다리꼴 넓이의 두 배이다.

$$C(i,j) = (\text{depth}[j] - \text{depth}[i])(A_i + A_j) = 2 \cdot \frac{A_i + A_j}{2}(\text{depth}[j] - \text{depth}[i])$$

- 따라서 경로상의 모든 점을 하나씩 확인하지 않고도, 넓이를 최소로 만드는 후보를 기하적으로 찾을 수 있다.

- 한 경로의 점들을 살펴보면, 최소 비용을 만드는 후보는 점들의 볼록 껍질 위에 놓인다.
- 그러므로 모든 조상을 매번 비교하는 대신, 현재 경로의 볼록 껍질만 관리하면 된다.
- 새 정점 x에 해당하는 점 P_x를 추가할 위치는, 껍질에서 연속한 점 P_i, P_{i+1}에 대해

$$\text{ccw}(P_i, P_{i+1}, P_x) > 0$$

  이 처음 성립하는 경계로 정한다.
- 볼록 껍질을 따라가며 외적의 부호가 한 번만 바뀌므로, 이 경계는 이분 탐색으로 찾을 수 있다.

R. 좀비고 고인물이 되고싶어!!

- 좌표가 $P_i = (A_i, \text{depth}[i])$ 일 때, 세 점의 방향은 다음 외적으로 계산한다.

$$\text{ccw}(P_i, P_{i+1}, P_x) = (A_{i+1} - A_i)(\text{depth}[x] - \text{depth}[i]) - (\text{depth}[i+1] - \text{depth}[i])(A_x - A_i)$$

- DFS로 내려가며 현재 정점의 점을 삽입하고, 그 정점의 자식들을 모두 처리한 뒤 삽입 전 상태로 되돌린다.
- 즉, 한 시점에는 현재 루트-정점 경로의 볼록 껍질만 유지한다. 자식 탐색이 끝나면 변경 기록을 롤백해 다음 형제도 올바른 경로의 껍질을 사용하게 한다.
- 정점마다 삽입 위치를 $O(\log N)$에 찾고 변경 사항을 롤백하므로, 전체 시간 복잡도는 $O(N \log N)$ 이다.

S. two convex sequences (hard)

`math` `geometry` `binary_search` `parametric_search` `convex_hull`

예상 난이도 – **Diamond IV**

- 본 대회 – 제출 횟수 8회 · 정답자 수 1명 · 정답률 12.50%
- Open Contest – 제출 횟수 0회 · 정답자 수 0명 · 정답률 —
- 최초 정답자 (본 대회): 2421 (1,600분)
- 최초 정답자 (Open Contest): 없음
- 출제자: `slow09`

S. two convex sequences (hard)

- hard에서는 M이 0이 아닐 수 있다. 볼록 조건을 이차 차분으로 정리하면 다음과 같다.

$$A_{i+1} - 2A_i + A_{i-1} \geq M,$$
$$B_{i+1} - 2B_i + B_{i-1} \leq -M.$$

- 이차 차분이 M인 보정 수열 P를 정의하자.

$$P_i = \frac{M(i-1)(i-2)}{2},$$
$$P_{i+1} - 2P_i + P_{i-1} = M.$$

S. two convex sequences (hard)

- $A'_i = A_i - P_i, B'_i = B_i + P_i$ 로 치환하면 다음을 얻는다.

$$A'_{i+1} - 2A'_i + A'_{i-1} \geq 0,$$
$$B'_{i+1} - 2B'_i + B'_{i-1} \leq 0.$$

- 따라서 A' 는 아래로 볼록하고 B' 는 위로 볼록해야 한다. 즉, normal과 같은 볼록 조건으로 바뀐다.

S. two convex sequences (hard)

- 각 원소의 범위에도 같은 보정을 적용한다.

$$AL_i - P_i \le A'_i \le AR_i - P_i,$$
$$BL_i + P_i \le B'_i \le BR_i + P_i.$$

- $P_1 = 0$ 이므로 첫 항은 변하지 않는다. 따라서 $A'_1 = B'_1 \in \mathbb{Z}$ 조건도 그대로 유지된다.
- 변환 전후의 수열은 서로 일대일로 대응한다. 보정한 범위에서 normal의 풀이를 그대로 적용하면 된다.
- 보정값은 커질 수 있으므로 64비트 정수로 저장하고, 외적과 선분 검사는 정수의 교차 곱으로 계산한다.
- 첫 항의 후보 범위는 변하지 않는다. 시간 복잡도는 $\mathcal{O}(N \log V)$, 공간 복잡도는 $\mathcal{O}(N)$ 이다.

EXTRA. 이 트리 뭐야

`ad_hoc` `trees` `binary_search` `parity`

예상 난이도 – **Unrated**

- 본 대회 – 제출 횟수 43회 · 정답자 수 1명 · 정답률 2.33%
- Open Contest – 제출 횟수 44회 · 정답자 수 0명 · 정답률 0.00%
- 최초 정답자(본 대회): 2421 (2,984분)
- 최초 정답자(Open Contest): 없음
- 출제자: k_san

EXTRA. 이 트리 뭐야

- 트리의 모든 간선은 기준점에서의 거리가 짝수인 정점과 홀수인 정점을 연결한다.
- 거리 k 인 정점의 존재 여부는 단조적이므로 가장 먼 정점을 이분 탐색으로 찾을 수 있다.
- 두 번의 이분 탐색으로 지름의 양 끝점 X, Y 와 길이 D 를 구한다.
- X 에서 거리 $\lfloor D/2 \rfloor$, Y 에서 거리 $\lceil D/2 \rceil$ 인 집합의 교점이 중심 O 이다.
- 중심에서 짝수 거리인 정점들을 질의로 모은다. 나머지는 홀수 거리인 정점들이다.
- 두 집합 중 작은 쪽에만 거리 1 을 질의하면 모든 간선을 복원할 수 있다.
- 전체 질의 수는 $20 + \lfloor (D+1)/4 \rfloor + \lfloor N/2 \rfloor \leq 395$ 이다.

EXTRA. 이 트리 뭐야

- $D \leq 400$ 이면 앞선 방법만으로 370 회 이하에 복원할 수 있다.
- $D > 400$ 이면 중심에서 거리가 4 의 배수인 정점들의 집합 S 를 구한다.
- S 의 정점들에 거리 1 을 질의하고, 그 이웃들의 집합을 B 라 하자.
- B 는 모두 중심에서 홀수 거리에 있으므로, 서로 연결된 정점이 없다.
- 따라서 B 를 제외한 정점들에만 거리 1 을 질의하면 된다. S 에는 이미 질의했다.
- 지름 경로에서 $|B| \geq (D-2)/2$ 가 보장된다. 레벨 질의는 $\lfloor (D+1)/8 \rfloor$ 회이다.
- 복원 질의는 350 회 이하이며, 중심 찾기까지 합쳐 최대 370 회이다.

EXTRA. 이 트리 뭐야

- 지름을 정확히 구하는 두 번째 이분 탐색의 비용을 줄여보자.
- 먼저 X 에서 거리 400 인 정점이 존재하는지 확인한다.
- 없다면 확인된 거리에서 4 씩 줄이며 처음으로 성공한 거리를 찾는다.
- 있다면 확인된 거리에서 4 씩 늘리며 마지막으로 성공한 거리를 찾는다.
- 성공한 거리와 정점을 R, Y 라 하면 $R \leq D \leq R + 3$ 이다.
- 경로 X–Y 의 중간 정점을 구하고, 2단계의 방법을 적용한다.
- 짧은 경우에는 짝수 거리 질의를 한 번 더 하여 중심의 오차를 보정한다.

- 공통 질의는 첫 이분 탐색 9회, 거리 400 확인 1회, 중간 정점 찾기 2회이다.
- 짧은 경우 거리를 4 줄일 때마다 탐색 질의는 하나 늘고, 짝수 거리 질의는 하나 줄어든다.
- 두 종류의 질의 합은 102회 이하이다. 전체 질의는 $9 + 1 + 2 + 102 + 250 = 364$회 이하이다.
- 긴 경우 추가로 성공한 탐색 횟수를 t라 하면 $R \geq 400 + 4t$이다.
- 추가 탐색은 $t + 1$회, 복원 질의는 $N - (R - 2)/2 + \lfloor (R + 1)/8 \rfloor$회 이하이다.
- 공통 질의 12회를 더해도 전체 364회 이하이므로 만점 조건을 만족한다.