2025 SCSC 알고리즘 대난투 풀이

Official Solutions

by

알고리즘 대난투 운영진

2025년 11월 29일

Staff

운영

- ohwphil
- sjhi00
- sjh1224
- sorohue
- terrasphere

- lycoris1600
- ksi4495
- sungjae0506
- ychangseok
- lumitt

검수

- leo020630
- yijw0930
- toycartoon
- swoon
- tony9402
- mujigae
- minsung05

문제		의도한 난이도	출제자
A	공통교육과정	**어려워요**	sjhi00
B	SCSC 동아리방 방문	**어려워요**	ohwphil
C	$K+1$의 배수	**어려워요**	sjhi00
D	짝수 길이의 짝수 합	**더어려워요**	sjhi00
E	쉬운 정렬 문제	**더어려워요**	terrasphere
F	직각 삼각형의 개수 2	**더어려워요**	sjhi00
G	봄 여름 가을 겨울	**더어려워요**	sjh1224
H	똥 피하기 게임	**더러워요**	sjh1224
I	마법사 루루와 마법의 숲	**진짜어려워요**	ohwphil
J	함수동상 그래프	**겁나어려워요**	sorohue

A. 공통교육과정

`implementation, string`

출제진 의도 – **어려워요**

- Rated Div. 제출 8 번, 정답 6 명 (정답률 75.000%)
- Rated Div. 처음 푼 사람: **Ai愛아이**, 1 분
- Open Contest 제출 86 번, 정답 76 명 (정답률 88.372%)
- Open Contest 처음 푼 사람: **flappybird**, 1 분
- 출제자: `sjhi00`

A. 공통교육과정

- 입력으로 받은 문자열이 `F`로 시작하면 `Foundation`, `C`로 시작하면 `Claves`, `V`로 시작하면 `Veritas`, `E`로 시작하면 `Exploration`을 출력하면 됩니다.
- Python에서는 `startswith()`, Java에서는 `startsWith()`, C++(C++20부터 가능)에서는 `starts_with()`를 사용할 수 있습니다.
- 문자열에서 첫 번째 문자열을 인덱싱(indexing)하는 방법(예시: 입력으로 주어지는 문자열이 `s`일 때 `s[0]`로 인덱싱)도 존재합니다.
- C++ 등의 언어에서는, 문자 하나만 입력받는 방법도 존재합니다.
- 시간 복잡도는 $\mathcal{O}(1)$ 입니다.

B. SCSC 동아리방 방문

`simulation`

출제진 의도 – **어려워요**

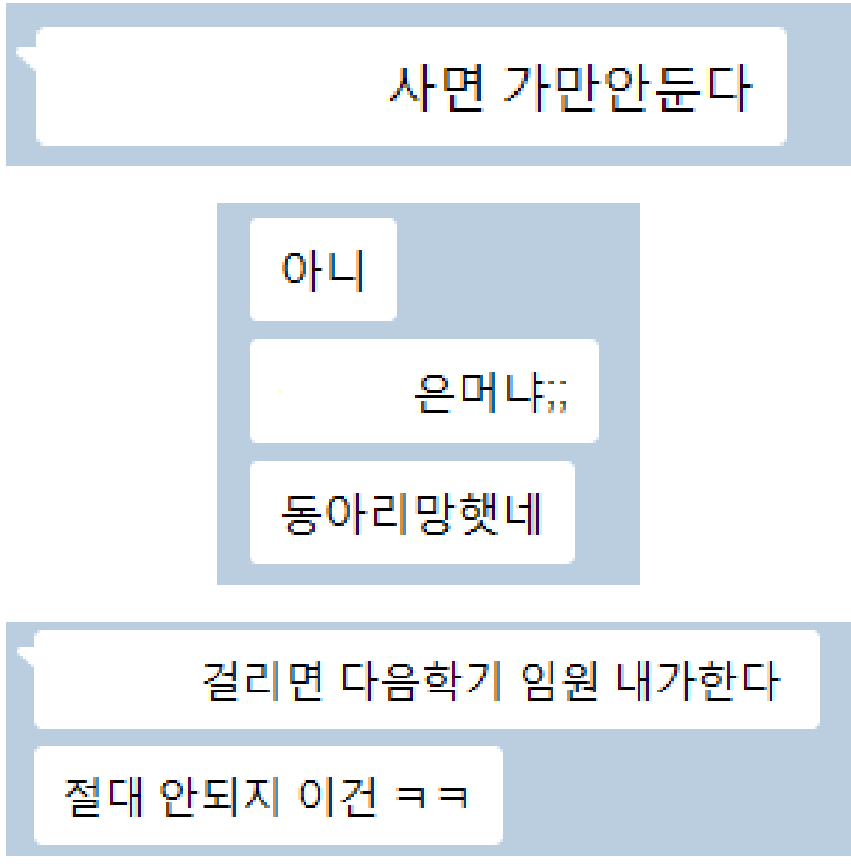
- Rated Div. 제출 6 번, 정답 6 명 (정답률 100.00%)
- Rated Div. 처음 푼 사람: **sjh1224 fan club**, 3분
- Open Contest 제출 54 번, 정답 39 명 (정답률 74.074%)
- Open Contest 처음 푼 사람: **flappybird**, 2분
- 출제자: `ohwphil`

B. SCSC 동아리방 방문

- 시뮬레이션을 통해 매번 가장 많이 남은 음료수의 개수가 몇 개이며, h_j 가 그만큼 많이 남아 있다면 l_j 번 음료수의 개수를 차감시키지 않고, 그렇지 않다면 l_j 번 음료수의 개수를 1 차감시키면 됩니다.
- 가장 많이 남은 음료수의 개수가 몇 개인지 구할 때 반복문을 사용하면 충분합니다.
- 시간복잡도는 $\mathcal{O}(NM)$ 입니다.

B. SCSC 동아리방 방문

- h_j 번 음료수가 들어올까 봐 경계하는 SCSC 부원들의 모습



C. $K+1$의 배수

`math`

출제진 의도 – **어려워요**

- Rated Div. 제출 10번, 정답 6명 (정답률 60.000%)
- Rated Div. 처음 푼 사람: **Ai愛아이**, 13분
- Open Contest 제출 84번, 정답 39명 (정답률 53.571%)
- Open Contest 처음 푼 사람: **hyperbolic**, 5분
- 출제자: `sjhi00`

C. $K+1$의 배수

- K가 짝수인 경우, 1부터 K까지의 정수를 선택하면,
  $1+2+\cdots+K=\frac{K(K+1)}{2}$이 되며, 이 값은 $K+1$의 배수입니다.
  즉, K가 짝수이면 항상 가능합니다.
- K가 홀수이고 $K=N$인 경우, $1+2+\cdots+K=\frac{K(K+1)}{2}$는
  $K+1$의 배수가 아니므로 불가능합니다.

C. $K+1$의 배수

- K가 홀수이고 $K < N$인 경우, $K+1$은 짝수입니다. 1부터 $K+1$까지의 정수 중에서 $\frac{K+1}{2}$를 제외하고 모두 선택하면 됩니다. 그러면 $(1+2+\cdots+K+(K+1)) - \frac{K+1}{2} = \frac{(K+1)(K+2)}{2} - \frac{K+1}{2} = \frac{(K+1)^2}{2}$이고, 이 값은 $K+1$의 배수입니다. 즉, 가능합니다.
- 따라서 K가 짝수이거나 $K < N$이면 가능하고, K가 홀수이고 $K = N$이면 불가능합니다.
- 시간 복잡도는 $\mathcal{O}(1)$입니다.

D. 짝수 길이의 짝수 합

`math`, `ad_hoc`

출제진 의도 – **더어려워요**

- Rated Div. 제출 43번, 정답 4명 (정답률 9.302%)
- Rated Div. 처음 푼 사람: **sjh1224 fan club**, 13분
- Unrated Div. 제출 3번, 정답 2명 (정답률 66.667%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 7분
- Open Contest 제출 37번, 정답 16명 (정답률 45.946%)
- Open Contest 처음 푼 사람: **hyperbolic**, 11분
- 출제자: `sjhi00`

D. 짝수 길이의 짝수 합

- a, b, c, d가 정수일 때, $a+b, c+d, a+b+c+d$ 중에서 적어도 하나는 짝수입니다.
- 만약 $a+b, c+d, a+b+c+d$가 전부 홀수라면, $a+b+c+d=(a+b)+(c+d)$는 두 홀수의 합이므로 짝수가 되어 모순이 발생합니다.
- 따라서 2번 쿼리에서는, $y-x+1 \geq 4$ 이면 항상 YES를 출력하면 되고, $y-x+1 \leq 3$ 이면 브루트포스로 확인하거나 조건문을 사용해서 확인하면 됩니다. 이때, 2번 쿼리 하나당 시간 복잡도는 $\mathcal{O}(1)$ 입니다.
- 입력으로 주어지는 문자열을 mutable(값의 변경이 가능한) 형태로 저장하면, 1번 쿼리 하나당 $\mathcal{O}(1)$의 시간복잡도로 수행할 수 있습니다.
- 전체 시간 복잡도는 $\mathcal{O}(N+Q)$ 입니다.

E. 쉬운 정렬 문제

`sorting`, `ad_hoc`

출제진 의도 – **더어려워요**

- Rated Div. 제출 15 번, 정답 5 명 (정답률 33.333%)
- Rated Div. 처음 푼 사람: **sjh1224 fan club**, 21 분
- Unrated Div. 제출 2 번, 정답 2 명 (정답률 100.00%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 9 분
- Open Contest 제출 33 번, 정답 18 명 (정답률 57.576%)
- Open Contest 처음 푼 사람: **hyperbolic**, 15 분
- 출제자: `terrasphere`

E. 쉬운 정렬 문제

- $i < j$ 이고, $A_i > A_j$ 인 쌍을 inversion 이라 칭하겠습니다.
- 정렬이 끝난 후에 이 둘의 순서는 바뀌므로 언젠가는 반드시 교환됨을 알 수 있습니다.
- 그러므로 정렬하기 위해서는 모든 inversion 쌍에 대해서 $A_i - A_j \leq K$ 여야 합니다.
- bubble sort 를 통해 위 조건을 만족하는 경우 항상 정렬 가능함을 보일 수 있습니다.

E. 쉬운 정렬 문제

- 이제 모든 inversion 쌍에 대해서 $A_i - A_j$ 의 최댓값을 구해야 합니다.
- 어떤 j 에 대해서 $A_i - A_j$ 를 최대로 만드는 A_i 는

$$\max_{i<j} A_i$$

  입니다. inversion 쌍이 없는 경우에 유의합니다.
- 따라서 A 배열을 왼쪽에서 오른쪽으로 순회하면서 이전까지 등장한 원소들의 최댓값을 관리하면 정렬에 필요한 K 의 최솟값을 구할 수 있습니다.
- 시간복잡도는 $O(N)$ 입니다.

F. 직각 삼각형의 개수 2

`bruteforcing`, `math`, `geometry`, `pythagoras`

출제진 의도 – **더어려워요**

- Rated Div. 제출 14 번, 정답 1 명 (정답률 7.143%)
- Rated Div. 처음 푼 사람: **sjh1224 fan club**, 95 분
- Unrated Div. 제출 12 번, 정답 1 명 (정답률 8.333%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 25 분
- Open Contest 제출 43 번, 정답 6 명 (정답률 13.953%)
- Open Contest 처음 푼 사람: **hyperbolic**, 42 분
- 출제자: `sjhi00`

F. 직각 삼각형의 개수 2

- 빗변이 아닌 두 변이 x 축 또는 y 축에 평행한 직각 삼각형의 경우, 직각에 해당하는 꼭짓점의 좌표를 (a, b) 라 하면, $x = a$ 위에 있는 점 중에서 (a, b) 가 아닌 점을 선택하고, $y = b$ 위에 있는 점 중에서 (a, b) 가 아닌 점을 선택할 수 있습니다.
- 즉, (a, b) 에 점이 존재할 경우, $x = a$ 위에 있는 점의 개수에서 1 을 뺀 값과, $y = b$ 위에 있는 점의 개수에서 1 을 뺀 값을 곱하고, 가능한 모든 (a, b) 에 대하여 이 값을 더하면 됩니다.
- 해당 아이디어는 BOJ 3000. 직각 삼각형에서도 쓰일 수 있습니다.
- 이제 빗변이 아닌 두 변이 x 축 또는 y 축에 평행한 경우가 아닌 직각 삼각형을 고려하면 됩니다.

F. 직각 삼각형의 개수 2

- 직각 삼각형에서 직각에 해당하는 꼭짓점의 좌표를 (a, b) 라 하고, 나머지 두 꼭짓점의 좌표를 (p, q), (r, s) 라 하겠습니다.
- 이때, 피타고라스의 정리에 의하여 $\left((p-a)^2+(q-b)^2\right)+\left((r-a)^2+(s-b)^2\right)=\left((p-r)^2+(q-s)^2\right)$ 입니다.
- 이 식을 정리하면, $(p-a)(r-a)+(q-b)(s-b)=0$ 을 얻습니다.
- $|p-a| \le N-1$ 이고 $|r-a| \le N-1$ 이므로, $|(q-b)(s-b)| = |-(p-a)(r-a)| = |p-a||r-a| \le (N-1)^2$ 입니다.
- 만약 $(q-b)(s-b)=0$ 이면, $(p-a)(r-a)=0$ 이므로, 빗변이 아닌 두 변이 x 축 또는 y 축에 평행한 경우가 되어 모순입니다. 즉, $(q-b)(s-b) \neq 0$ 입니다.

F. 직각 삼각형의 개수 2

- $q-b$와 $s-b$가 정수이므로, $|q-b|$와 $|s-b|$는 $(N-1)^2$ 이하의 정수입니다.
- $A=p-a, B=r-a, C=q-b, D=s-b$로 치환해서 생각할 때, $1 \le |A|, |B| \le N-1$이고, $AB+CD=0$을 만족하는 정수 순서쌍 (A,B,C,D)가 그다지 많지 않음을 알 수 있습니다.
- 순서쌍 (a,b)와 $(p-a,q-b,r-a,s-b)$가 결정되면, 두 꼭짓점의 좌표 (p,q), (r,s)도 자동 결정됩니다.
- 따라서 가능한 모든 순서쌍 $(p-a,q-b,r-a,s-b)$를 찾은 다음, 가능한 (a,b)에 대해서 브루트포스로 확인해보면 제한 시간 내에 답을 구할 수 있습니다.

F. 직각 삼각형의 개수 2

- 이제부터는 이 방법이 제한 시간 내에 돌아가는 이유, 즉 시간 복잡도에 대한 자세한 설명입니다. 문제 풀이에는 직접적인 도움이 되지는 않는 내용입니다.
- 양의 정수 k에 대하여, k의 양의 약수의 개수를 $\sigma_0(k)$라고 하겠습니다.
- $p-a$와 $r-a$의 값을 절댓값이 $N-1$ 이하인 정수 범위에서 선택하면, 가능한 순서쌍 $(q-b, s-b)$의 개수는 $2\sigma_0(|(p-a)(r-a)|)$가 됩니다.
- 즉, 가능한 순서쌍 $(p-a, q-b, r-a, s-b)$의 개수는 대략 $\sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\sigma_0(ij)$의 상수배입니다.

F. 직각 삼각형의 개수 2

- 한편, 양의 정수 x, y에 대하여 $\sigma_0(xy) \leq \sigma_0(x)\sigma_0(y)$ 입니다.
- 또한, 양의 정수 M에 대하여 $\sum_{i=1}^{M} \sigma_0(i) = \sum_{i=1}^{M} \lfloor \frac{M}{i} \rfloor = \mathcal{O}(M \log M)$ 입니다.
- 따라서 $\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sigma_0(ij) \leq \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_0(i)\sigma_0(j) = \left(\sum_{i=1}^{N} \sigma_0(i) \right)^2 = \mathcal{O}\left(N^2 (\log N)^2 \right)$ 입니다.
- 가능한 모든 (최대 NM 개의) (a, b)에 대해서 확인하게 되므로, 시간 복잡도는 대략 $\mathcal{O}\left(N^3 (\log N)^2 M \right)$ 입니다.

F. 직각 삼각형의 개수 2

- 비록 $(p-a, q-b, r-a, s-b)$ 에서 부호 $(+, -)$ 를 고려해야 하므로 시간복잡도에 비교적 큰 값의 상수가 붙을 수는 있지만, $N \leq 5$ 이고, $p-a$ 와 $r-a$ 의 절댓값이 $N-1$ 이하이며, 실제로는 (a, b) 에서 a 의 값에 따라 가능한 $p-a, r-a$ 의 값의 개수가 달라질 수 있으므로, 생각보다 많은 계산량을 요구하지는 않습니다.
- 실제로 $N = 5$ 이고, a 가 고정된 정수 $(1 \leq a \leq N)$ 일 때, $(p-a)(r-a) + (q-b)(s-b) = 0, (q-b)(s-b) \neq 0, 1 \leq p, r \leq N$ 을 만족하는 정수 순서쌍 (b, p, q, r, s) 의 개수는 416 개입니다. (p, q) 와 (r, s) 의 대칭성을 고려하면 208 개만 고려해도 충분합니다.
- $208 \times 200\,000 \approx 4 \times 10^7$ 이므로, 제한 시간 내에 답을 구할 수 있습니다.

G. 봄 여름 가을 겨울

`tree_set`

출제진 의도 – **더어려워요**

- Rated Div. 제출 3번, 정답 1명 (정답률 33.333%)
- Rated Div. 처음 푼 사람: **sjh1224 fan club**, 176분
- Unrated Div. 제출 7번, 정답 2명 (정답률 28.571%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 34분
- Open Contest 제출 9번, 정답 5명 (정답률 55.556%)
- Open Contest 처음 푼 사람: **yaksha**, 41분
- 출제자: `sjh1224`

G. 봄 여름 가을 겨울

- i 번 달의 계절을 구하는 방법을 알아봅시다.
- $A_j \neq A_{j+1}, j < i$ 인 최대 j 를 p 라고 합시다.
- p 가 존재하지 않는다면, 1 번 달의 계절의 정의에 의해 i 번 달은 봄입니다.
- p 가 존재한다면, i 번 달의 계절은 다음과 같은 방식으로 결정할 수 있습니다.
  - $p = i - 1, A_p < A_{p+1}$ 이라면 i 번 달의 계절은 여름입니다.
  - $p = i - 1, A_p > A_{p+1}$ 이라면 i 번 달의 계절은 겨울입니다.
  - $p < i - 1, A_p < A_{p+1}$ 이라면 i 번 달의 계절은 가을입니다.
  - $p < i - 1, A_p > A_{p+1}$ 이라면 i 번 달의 계절은 봄입니다.

G. 봄 여름 가을 겨울

- 따라서 p를 구하기만 하면 $\mathcal{O}(1)$ 시간에 2번 쿼리를 처리할 수 있습니다.
- $S = \{j : A_j \neq A_{j+1}\}$을 std::set이나 Range Max Segment Tree로 관리해주면, 업데이트 쿼리와 p 구하기를 모두 $\mathcal{O}(\log N)$ 시간에 처리할 수 있습니다.
- 따라서 전체 쿼리를 $\mathcal{O}(N + Q \log N)$ 또는 $\mathcal{O}((N + Q) \log N)$ 시간에 처리할 수 있습니다.

H. 똥 피하기 게임

`graph_traversal`

출제진 의도 – **더러워요**

- Rated Div. 제출 13번, 정답 0명 (정답률 0.000%)
- Rated Div. 처음 푼 사람: **—**, —분
- Unrated Div. 제출 9번, 정답 2명 (정답률 22.222%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 103분
- Open Contest 제출 35번, 정답 7명 (정답률 20.000%)
- Open Contest 처음 푼 사람: **mingyu332**, 50분
- 출제자: `sjh1224`

H. 똥 피하기 게임

- 똥이 1 초마다 1 칸씩 떨어지는 게 아니라, 캐릭터가 1 초마다 1 칸씩 올라간다고 생각해봅시다.
- 즉, 모든 똥은 정지해 있으며, 캐릭터는 1 초마다 (i, j) 에서 $(i-1 \pmod n, j-1)$ 또는 $(i-1 \pmod n, j+1)$ 으로 이동해야 합니다.
- 격자의 각 빈칸에서 캐릭터가 이동할 수 있는 다른 빈칸으로 간선을 이어 그래프를 만들어봅시다.
- 어떤 정점에서 어떤 사이클로 도달하는 경로가 존재하는 것과 그 정점에서 시작해 무한히 게임이 끝나지 않게 할 수 있는 것이 동치입니다.
- 따라서 맨 아래 행의 각 정점에서 어떤 사이클로 도달하는 경로가 존재하는지 판정하면 문제를 풀 수 있습니다

H. 똥 피하기 게임

- 맨 아래 행의 각 정점마다 한 번씩 DFS를 하여 판정해주면 $\mathcal{O}(NM^2)$ 시간에 문제를 풀 수 있습니다.
- 더 효율적인 구현을 생각해봅시다.
- 그래프의 모든 간선을 반대 방향으로 뒤집으면, '그 정점에서 어떤 사이클로 도달하는 경로가 존재하는지'가 '어떤 사이클에서 그 정점으로 도달하는 경로가 존재하는지'로 바뀝니다.
- 위상 정렬을 할 때처럼 indegree가 0인 정점을 하나씩 방문하며 지워주는 순회를 하면, 방문하지 못하는 정점이 정확히 '어떤 사이클에서 도달하는 경로가 존재하는 정점'과 같습니다.
- 따라서 $\mathcal{O}(NM)$ 시간에 문제를 풀 수 있습니다.

I. 마법사 루루와 마법의 숲

`graph_theory`, `tree`, `graph_traversal`
출제진 의도 – **진짜어려워요**

- Rated Div. 제출 0번, 정답 0명 (정답률 0.000%)
- Rated Div. 처음 푼 사람: **—**, —분
- Unrated Div. 제출 3번, 정답 2명 (정답률 66.667%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 130분
- Open Contest 제출 5번, 정답 3명 (정답률 60.000%)
- Open Contest 처음 푼 사람: **mingyu332**, 27분
- 출제자: `ohwphil`

- 1 번째 실행: 숲으로부터 트리 만들기
- 숲과 트리의 정점을 바꾸는 순열 A 가 $N+1$ 번 정점의 번호는 바꾸지 않는다는 점에 주목해 봅시다.
- 일단 가장 직관적으로 드는 생각은, 숲 F 에 있는 각 트리의 특별한 간선의 양 끝 정점 중 하나와, $N+1$ 번 정점을 잇는 방법을 생각할 수 있습니다.
- 그렇다면 양 끝 정점 중 어떤 것을 이으면 될까요? 정점의 번호가 섞여도 영향을 받지 않는 "서브트리의 크기"를 활용할 수 있습니다.
- 트리를 특별한 간선을 기준으로 끊었을 때, 크지 않은 서브트리에 속한 정점과 $N+1$ 번 정점을 이어 봅시다.

- 2 번째 실행: 트리로부터 숲 복원하기
- 그렇다면 숲을 복원할 때, $N+1$ 번 정점과 연결된 간선을 모두 끊으면 기존의 트리는 복원되고, 특별한 간선만 복원하면 됩니다.
- 기존에 $N+1$ 번 정점과 연결되었던 정점과 연결된 간선 중에서, 어떤 것이 특별한 간선일까요?
- 1 번째 실행에서 u 번 정점과 v 번 정점을 잇는 특별한 간선이 있었고, u 번 정점 쪽에 속한 서브트리가 크지 않아서 u 번 정점과 $N+1$ 번 정점을 이었다고 합시다.
- u 번 정점과 v 번 정점이 숲 F 에서 속했던 트리의 정점이 M 개였다고 한다면, 특별한 간선을 끊은 후 u 번 정점이 속한 서브트리에는 $\lfloor \frac{M}{2} \rfloor$ 개 이하의 정점이 있었고, v 번 정점이 속한 서브트리에는 $\lceil \frac{M}{2} \rceil$ 개 이상의 정점이 있었습니다.

- 2 번째 실행: 트리로부터 숲 복원하기
- 이제 트리 T' 에서 A_u 번 정점과 인접한 정점 중에서 어떤 것이 A_v 일까요?
- A_u 번 정점과 $N+1$ 번 정점을 연결하는 간선을 끊었을 때, A_u 번 정점이 속한 트리에는 M 개의 정점이 있습니다.
- 이제 이 트리를 A_u 번 정점을 루트로 하는 트리로 생각해 보면, A_u 번 정점과 인접하면서 서브트리의 크기가 $\lceil \frac{M}{2} \rceil$ 이상인 정점은 유일하게 존재합니다.
- 따라서 이 정점이 A_v 이고, 우리는 특별한 간선을 찾아내어 성재와 오필을 구출해낼 수 있습니다.
- 모든 과정의 시간복잡도는 $\mathcal{O}(N)$ 입니다.

J. 함수동상 그래프

`functional_graph`, `dp_tree`

출제진 의도 – **겁나어려워요**

- Rated Div. 제출 0 번, 정답 0 명 (정답률 0.000%)
- Rated Div. 처음 푼 사람: **—** , — 분
- Unrated Div. 제출 1 번, 정답 1 명 (정답률 100.000%)
- Unrated Div. 처음 푼 사람: **jjang36524**, 161 분
- Open Contest 제출 5 번, 정답 2 명 (정답률 40.000%)
- Open Contest 처음 푼 사람: **jeoffrey0522**, 48 분
- 출제자: `sorohue`

J. 함수동상 그래프

- 편의 상 동상이 올려진 정점의 집합을 **정답 집합**으로 칭하겠습니다.
- 그래프가 여러 개의 컴포넌트로 나뉘어 있는 경우, 각 컴포넌트에 대한 답을 곱해 전체 그래프에서의 답을 얻을 수 있습니다.
- 단일 컴포넌트에서 답을 구해 봅시다.

J. 함수동상 그래프

- 주어지는 그래프는 **함수형 그래프**입니다. 함수형 그래프의 한 컴포넌트는 정확히 하나의 단순 사이클을 갖습니다. 게다가 사이클에 포함되지 않은 모든 간선은 사이클로 향하는 방향을 갖습니다.
- 그래프에서 사이클을 이루는 간선들을 제거하면 남는 정점들은 포레스트를 이룹니다. 이때 사이클에 속해 있던 정점들을 각 트리의 루트로 설정하면, 모든 간선은 해당 간선이 속한 트리의 루트 쪽을 향합니다.

J. 함수동상 그래프

- V 개의 정점으로 이루어진 사이클 위에 S 개의 동상이 올려져 있는 상황을 고려합시다. 사이클 위의 동상들은 행동을 통해 사이클 안에서 순서를 유지하면서 사이클 상에서의 위치를 임의로 결정할 수 있으므로, 가능한 정답 집합의 수는 $\binom{V}{S}$ 입니다.
- 사이클에 가지가 붙어 있는 경우, 가지에서 몇 개의 동상이 사이클로 옮겨질 수 있습니다. 그러한 경우를 모두 고려해, 사이클 위의 최종 동상 개수 별로 가지들만 남겼을 때의 정답 집합 수를 세어 사이클만 남겼을 때의 정답 집합 수와 곱해 이를 모두 합하면 한 컴포넌트에 대한 답을 구할 수 있습니다.

J. 함수동상 그래프

- 전술했듯 각각의 가지는 루트 있는 트리입니다.
- 트리 DP가 하고 싶어집니다...

J. 함수동상 그래프

- $d[v][k]$ 를 v 번 정점을 루트로 하는 서브트리에 k 개의 동상을 남기는 경우의 수로 정의합니다.
- $d[v]$ 를 다항식으로 생각할 수 있습니다. $d[v][k]$ 가 $d[v]$ 의 k 차항의 계수가 되는 식입니다.
  - 생성 함수가 무엇인지 알고 있다면 그걸로 이해해도 좋습니다. 몰라도 문제를 해결하는 데 지장은 없습니다.

J. 함수동상 그래프

- v 번 정점에게 t 개의 자식 정점 $w_1, w_2, \cdots, w_t$ 가 있다고 합시다. 다항식 $\prod_{i=1}^{t} d[w_i]$ 가 자식 서브트리들의 합집합에 대한 값을 표현함을 알 수 있습니다.
- 여기서 v 번 정점에 동상을 남기는 경우를 고려합니다. 서브트리들에서 빠져나간 동상 중 최대 하나만 v 번 정점에 남을 수 있기 때문에, 이는 간단히 직전의 다항식에 $x+1$ 을 곱하는 것으로 표현할 수 있습니다.
- 이때 v 번 정점에 동상이 있는지를 확인해야 합니다. 위의 계산 결과로 나오는 다항식의 최고차항은 동상을 서브트리 밖으로 내보내지 않으면서 v 번 정점에 동상이 존재하는 경우의 수에 해당해, v 번 정점에 동상이 없다면 불가능한 경우가 됩니다. 이 경우 간단히 최고차항을 지워주기만 하면 됩니다.

J. 함수동상 그래프

- 그래서 서브트리끼리 합치는 과정을 다항식 곱셈으로 나타낼 수 있습니다.
- 다항식 곱셈을 나이브하게 구현하면 곱셈 한 번에 제곱 시간 복잡도가 걸릴 거 같은데...
- FFT가 하고 싶어집니다...

- **라는 생각을 떨쳐 버리고 그냥 나이브하게 다항식 곱셈을 구현해도 됩니다!**
- 크기가 각각 A, B 인 두 서브트리를 합치는 데에 $\mathcal{O}(AB)$ 번의 연산이 필요한 것은 사실입니다. 그러나 트리의 기묘하고 놀라운 성질 덕분에, 트리 전체에 대한 DP를 수행하는 시간 복잡도는 $\mathcal{O}\left(N^2\right)$ **이 됩니다.**
  - 간단히 설명하면, 두 서브트리를 크기의 곱에 합치는 행위는, 각 서브트리에서 정점을 하나씩 고르는 모든 경우의 수를 시도해 보는 것과 같은 시간 복잡도를 갖습니다. 후자의 행위를 트리 전체에서 하면 결과적으로 트리 전체에서 임의로 두 정점을 고르는 모든 경우의 수를 한 번씩 시도한 꼴이 돼서, $\mathcal{O}\left(N^2\right)$ 의 시간 복잡도가 걸림을 알 수 있습니다.
  - KOI의 유명한 문제 이름을 따서 이런 성질을 검은 돌 트릭이라고도 부릅니다.

J. 함수동상 그래프

- 이제 각 가지에 대해서 필요한 값을 구했으니, 사이클과 합쳐주면 끝입니다.
- 트리를 합칠 때와 비슷한 방식으로 다항식 곱셈의 형태로 컴포넌트에서의 정답 집합의 수를 계산할 수 있음을 알 수 있습니다.
- 근데 이번에는 트리 모양이 아닌데요? 이번에는 진짜 FFT죠?????

- 아니요. 다항식을 곱해 합치는 과정을 트리처럼 표현할 수 있어서, 직전과 동일한 이유로 나이브한 다항식 곱셈을 이용했을 때의 시간 복잡도가 $\mathcal{O}\left(N^2\right)$ 으로 보장됩니다.
- 따라서 총 시간 복잡도 $\mathcal{O}\left(N^2\right)$ 에 문제를 해결할 수 있습니다!

J. 함수동상 그래프

Challenge

1. $N \leq \mathbf{1\,000\,000}, K \leq 4\,000$ 에서 같은 문제를 풀어 보세요.
2. 0 이상 $1\,000\,000\,006$ 이하의 모든 정수 x 에 대해,
   답이 $x \mod 1\,000\,000\,007$ 인 테스트 케이스를 만들 수 있나요?