

240

신촌지역 대학생 프로그래밍 동아리 연합 여름 대회 | Main Contest Solution Writeup

SUAPC 2026 Summer | 본대회 풀이 및 해설

주최

ICPC Sinchon

후원



Dify



생능북스

dcamp

cenix820 cubic jtw7913 nanoka ramma solved.ac 김태우 김태윤(ystaeyoon113) 안녕원팬 이재열 임예찬

문제 목록 및 난이도 요약

A 수앞C	sorohue	Bronze III	H README	woohyun_jng	Gold III
B 분탕의 신 안녕원	lunarlity	Platinum II	I 책 정렬	pyb1031	Diamond IV
C 공 넣기	pk661	Gold III	J 트리 뽑아내기	realpsdoingdamyoo	Platinum III
D Regulating Bracket Sequence	hibye1217	Platinum II	K 술 푸는 사람	cohenpf	Platinum I
E K분 그래프 3	jms020820	Diamond V	L 정투상도	cohenpf	Gold III
F 122333	shinhuichan	Silver I	M 쌀 재배하기와 쿼리	hyperbolic	Platinum III
G 쌀 가져가기와 쿼리	hyperbolic	Diamond III	N 매직 쌀 2	annyeong1	-

A. 수앞C

의도된 난이도 **Bronze III**

#string

출제자

정시우 (sorohue)

제출 32번, 정답 25팀

정답률: 81.25%

처음 풀 팀

(공식) Don't Die Yet / 2분
(비공식) 네! 다음은 재밌고 신박한
팀명의 목록입니다. / 1분

A. 수앞C

```
print(sum(a=='C' and b.isdigit() for a,b in zip(S,S[1:])))
```



B. 분탕의 신 안녕원

의도된 난이도 **Platinum II**

#constructive #ad_hoc #trees #invariant

출제자

송주현 (lunarlity)

제출 16번, 정답 2팀

정답률: 12.50%

처음 풀 팀

(공식) 없음 / -

(비공식) MIKUDAYO / 151분

B. 분탕의 신 안녕원

- 검은색, 흰색 대신 각각 0, 1로 표기하자.
- $1 \leq X \leq 10^9 < 2^{30}$ 이므로, X 를 전달하는 문제는 곧 30비트를 전달하는 문제로 생각할 수 있다.
- 먼저 $x - y - z$ 형태의 구조를 생각하자. 만약 x 에는 연산이 적용되지 않는다고 하면, $y \oplus z$ 는 불변량이다.
- 따라서 x 에 연산이 적용되었는지만 알 수 있다면, $y \oplus z$ 를 이용해 1비트를 전달할 수 있다.

B. 분탕의 신 안녕원

- 이를 위해서 $x - w - v$ 라는 가지를 하나 더 붙이자.
- x 에 연산이 적용되지 않았다면, w 나 v 에 연산을 몇 번 하더라도 $w \oplus v$ 는 유지된다.
- 반면 x 에 연산이 적용되면 w 와 y 만 뒤집히므로, 결과적으로 $w \oplus v$ 의 값이 1만큼 뒤집힌다.
- 즉 $w \oplus v$ 를 보면 x 에 연산이 적용되었는지를 알 수 있다.
- 이를 종합하면 $w \oplus v \oplus y \oplus z$ 자체가 불변량이 된다.
- 즉 x, y, z, w, v 의 총 5개 정점으로 1비트를 전달할 수 있다.

B. 분탕의 신 안녕원

- 이제 이를 30개 이어 붙이면 된다.
- $x_1 - x_2 - \dots - x_{30}$ 을 경로로 만들고, 각 x_i 에 y_i, z_i, w_i, v_i 를 위와 같은 형태로 붙이자.
- 그러면 각 i 에 대해 $y_i \oplus z_i \oplus w_i \oplus v_i$ 는 불변량이 되므로, 이를 이용해 30비트를 전달할 수 있다.
- 다만 정점 라벨링이 사라지는 경우를 처리해야 한다.
- 이를 위해 x_1 쪽에 의미 없는 정점 하나를 더 붙여 시작점을 구분해 주면, 전체 경로의 방향이 정해진다.
- 따라서 Bob은 어느 정점이 $x_1, \dots, x_{30}$ 에 해당하는지 복원할 수 있고, 각 위치에서 위 불변량을 읽어 30비트를 복원할 수 있다. $N = 151$ 이다.

C. 공 넣기

의도된 난이도 **Gold III**

#constructive

출제자

손재원 (pk661)

제출 80번, 정답 24팀

정답률: 30.00%

처음 푼 팀

(공식) 수아가 만든 컴퓨터는

수아피씨 / 15분

(비공식) intiger / 13분

C. 공 넣기

- 인접한 두 상자는 같은 시행에서 선택할 수 없으므로, 적어도 $a_i + a_{i+1}$ 번의 시행이 필요하다.
- 그러므로 최소 시행 횟수 $K \geq \max_{1 \leq i \leq N} (a_i + a_{i+1})$ 이다.
- 한편, $K = \max_{1 \leq i \leq N} (a_i + a_{i+1})$ 회에 구성이 가능하다.
 - 홀수 번째 상자에는 $1, 2, \dots, a_i$ 번째 시행에,
 - 짝수 번째 상자에는 $K - a_i + 1, \dots, K - 1, K$ 번째 시행에 공을 넣어주면 된다.

D. Regulating Bracket Sequence

의도된 난이도 **Platinum II**

#ad_hoc #string #kmp

출제자
hibye1217

제출 12번, 정답 1팀
정답률: 8.33%

처음 풀 팀
(공식) 없음 / -
(비공식) 1 MiB / 284분

D. Regulating Bracket Sequence

- 출제자 하씨보다 더 아름답게 풀 이름의 하씨의 풀이를 소개합니다.
- 문제에서 주어진 연산 중 $(A)B$ 를 $B(A)$ 로 만드는 건 그냥 문자열을 Cyclic Shift한 것과 같습니다.
- 그럼 $(A)B$ 를 $(B)A$ 로 만드는 것도 Cyclic Shift로 표현할 수 있을까요?

D. Regulating Bracket Sequence

- $A(B)$ 를 $B)A$ (로 만든 뒤 $)$ (를 다시 $()$ 쌍으로 맞춰주면 됩니다!
- 따라서, 쌍만 맞는다면 (와)를 구분할 필요가 없습니다.
- 따라서 각 쌍에 대해 해당 쌍의 괄호 종류와 거리만 저장한 배열을 만든 뒤, Cyclic Shift를 잘 해서 두 배열이 같아지는지 확인하면 됩니다.
- 이는 KMP로 간단히 해결할 수 있습니다.

D. Regulating Bracket Sequence

- 출제자는 어떻게 풀었냐고요? 트리 동형 사상이요.

E. K분 그래프 3

의도된 난이도 **Diamond V**

#graph #cycle basis #math #Inclusion Exclusion

출제자

조문성 (whans)

제출 9번, 정답 2팀

정답률: 22.22%

처음 풀 팀

(공식) 中國人 / 291분

(비공식) intiger / 212분

E. K분 그래프 3

- 정점 1을 루트로 잡고, 아무 DFS 트리를 구성합니다. 루트부터 v 까지의 거리를 D_v 라 하겠습니다.
- 간선 (u, v, w) 마다 $X_{u,v} = D_u + w - D_v$ 를 정의하고, 모든 $X_{u,v}$ 의 절댓값의 최대공약수를 G 라 하겠습니다.
- 임의의 $u \rightarrow v$ walk가 지나는 간선들의 X 값을 $X_1, \dots, X_t$ 라 하면, 그 가중치 합 W 는 $W = D_v - D_u + \sum_{i=1}^t X_i$ 로 나타낼 수 있습니다. 따라서 항상 $W \equiv D_v - D_u \pmod{G}$ 입니다.
- 또한 닫힌 walk를 덧붙여 가중치 합을 G 의 배수 단위로 변화시킬 수 있으므로, 가능한 가중치 합은 $D_v - D_u + tG$ 꼴입니다.

E. K분 그래프 3

- 가중치가 k 의 배수인 walk가 존재하려면 어떤 정수 t 에 대해 $D_v - D_u + tG \equiv 0 \pmod{k}$ 를 만족해야 합니다.
- 이는 일차 합동식 $tG \equiv -(D_v - D_u) \pmod{k}$ 의 해 존재 여부와 같으므로, 필요충분조건은 $\gcd(G, k) \mid D_v - D_u$ 입니다.
- $H = \gcd(G, |D_v - D_u|)$ 라 두면 조건은 $\gcd(G, k) \mid H$ 와 같습니다. 즉, 쿼리의 답은 u, v 자체가 아니라 H 에만 의존합니다.
- 따라서 G 의 모든 약수 H 에 대한 답을 미리 계산하면 각 쿼리를 빠르게 처리할 수 있습니다.

E. K분 그래프 3

- 이제 쿼리에서 $H = \gcd(G, |D_v - D_u|)$ 만 알면 되고, 다음 값을 구하면 됩니다.
 $|\{1 \leq k \leq K : \gcd(G, k) \mid H\}|$

$$G = \prod_{i=1}^r p_i^{a_i}, \quad H = \prod_{i=1}^r p_i^{b_i} \quad (0 \leq b_i \leq a_i)$$

라고 합시다.

- $\gcd(G, k) \mid H$ 가 성립하려면, $b_i < a_i$ 인 소수 p_i 에 대해서 $k \not\equiv 0 \pmod{p_i^{b_i+1}}$ 이어야 합니다.
- 즉, $R_i = p_i^{b_i+1}$ 라 할 때 1부터 K 까지의 정수 중 R_i 의 배수들을 제외하는 문제가 됩니다.

E. K분 그래프 3

- $I = \{i \mid b_i < a_i\}$ 라 하겠습니다.
- 각 $i \in I$ 에 대해 $R_i = p_i^{b_i+1}$ 의 배수는 조건을 만족하지 않습니다.
- 여러 R_i 의 배수인 수는 여러 번 제외되므로 포함배제를 적용합니다.
- 서로 다른 R_i 들은 서로 다른 소수의 거듭제곱이므로 서로소입니다. 따라서

$$\text{Ans}(H) = \sum_{S \subseteq I} (-1)^{|S|} \left\lfloor \frac{K}{\prod_{i \in S} R_i} \right\rfloor$$

입니다.

E. K분 그래프 3

- G 를 소인수분해하고 모든 약수를 구한 뒤, 각 H 에 대해 위 포함배제를 수행하여 $\text{Ans}(H)$ 를 미리 계산합니다.
- 이후 각 쿼리 (u, v) 에 대해서는 $H = \gcd(G, |D_v - D_u|)$ 를 구하고 $\text{Ans}(H)$ 를 출력하면 됩니다.
- $G = 0$ 이라면 모든 간선에서 $D_u + w - D_v = 0$ 입니다. 강한 연결성과 음이 아닌 가중치 조건에 의해 모든 간선의 가중치가 0이므로 모든 $1 \leq k \leq K$ 가 가능하고, 모든 쿼리의 답은 K 입니다.

E. K분 그래프 3

- $G \leq N \cdot \max w \leq 2 \cdot 10^{14}$, $\sqrt{G} \leq 2 \cdot 10^7$ 이므로 G 의 소인수분해는 직접 나누는 식으로 충분히 빠르게 할 수 있습니다.
- $\tau(G)$ 를 G 의 약수의 개수, $\omega(G)$ 를 서로 다른 소인수의 개수라 하겠습니다.
- $G = \prod p_i^{a_i}$ 라 하면, 모든 약수에 대한 포함배제의 총 연산 횟수는 $\sum_{h|G} \omega(h) 2^{\omega(h)} \leq \omega(G) \prod (2a_i + 1)$ 입니다.
- $G \leq 2 \cdot 10^{14}$ 에서 $\omega(G) \leq 12$, $\tau(G) \leq 20160$, $\prod (2a_i + 1) \leq 2657205$ 이므로 충분히 빠르게 처리할 수 있습니다.
- 전체 시간복잡도는 $O(N + M + \sqrt{G} + \omega(G) \prod (2a_i + 1) + Q \log \tau(G))$ 입니다.

F. 122333

의도된 난이도 **Silver I**

#greedy #implement

출제자

신희찬(shinhuichan)

제출 45번, 정답 25팀

정답률: 55.56%

처음 풀 팀

(공식) 中國人 / 8분

(비공식) Credits EXPERT / 6분

F. 122333

- 이 문제는 그리디하게 해결 할 수 있습니다.
- 원소를 차례대로 봅니다.
- 현재까지 완료할 수 있는 최대 단계를 저장합니다.
- 그 단계에서 원소 위치의 최댓값의 가능한 최솟값도 저장합니다.
- 최대 단계를 업데이트 할 수 있는지 확인합니다.
- 이 과정을 반복하면 답이 나옵니다.
- $O(N)$ 시간복잡도로 해결 가능합니다.

G. 쌀 가져가기와 쿼리

의도된 난이도 **Diamond III**

#flow #duality #knapsack #segment tree

출제자

임병찬 (hyperbolic)

제출 6번, 정답 2팀

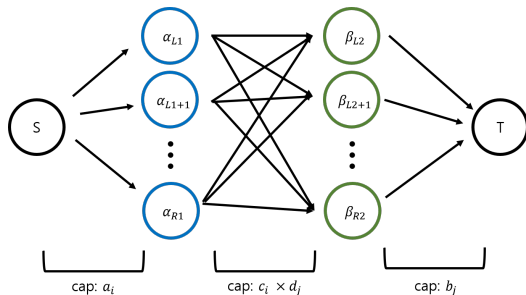
정답률: 33.33%

처음 풀 팀

(공식) 中國人 / 212분

(비공식) intiger / 269분

G. 쌀 가져가기와 쿼리

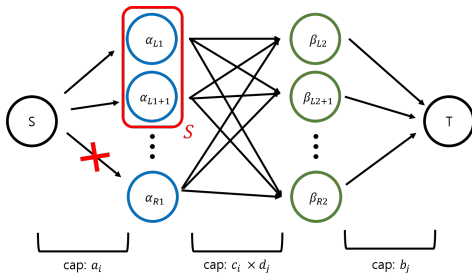


- 각 쿼리의 답은 위의 그래프를 만든 후 maximum flow를 계산하면 구할 수 있습니다.

G. 쌀 가져가기와 쿼리

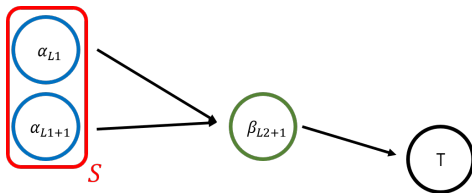
- 하지만 maximum flow의 계산은 시간이 굉장히 오래걸립니다. $Q = 1$ 을 가정하더라도 문제를 쉽게 풀 수 없어보입니다.
- maximum flow와 minimum cut은 언제나 같기 때문에, maximum flow 대신 minimum cut을 계산해보는 방식을 생각해 볼 수 있습니다.
- minimum cut은 어떻게 구할 수 있을까요?

G. 쌀 가져가기와 쿼리



- 먼저, 왼쪽 노드 집합의 부분집합 S 를 설정해봅시다. 그 후, S 에 속하지 않은 모든 왼쪽 노드와 source사이의 간선을 끊어봅시다.
- 모든 간선을 끊는데 총 $\sum_{i \notin S} a_i$ 의 cost가 필요합니다.

G. 쌀 가져가기와 쿼리



- 오른쪽 노드에 대해서 생각해봅시다. S 에 속하지 않은 왼쪽 노드들은 이미 source와 연결이 끊어졌기 때문에, 이 간선들은 자를 필요가 없습니다.
- 그러므로, S 에 속한 모든 왼쪽노드와 연결된 간선들을 모두 끊거나, sink와 연결된 간선을 끊거나 둘 중 하나 이상의 행동을 해야합니다. 둘다 하지 않은 경우, source와 sink가 연결되기 때문입니다.

G. 쌀 가져가기와 쿼리

- 저희가 구해야 하는건 minimum cut입니다. 그러므로, 둘 중 하나 이상의 행동을 해야한다고 하였지만 사실은 반드시 둘 중 하나의 행동만을 해야합니다.
- 결론적으로, source와 sink사이에 연결을 끊기 위해서는 $\sum_{j=L_2}^{R_2} \min(\sum_{i \in S} c_i \times d_j, b_j)$ 의 cost가 필요합니다.
- 앞에서 S 에 속하지 않은 왼쪽 노드와의 간선을 자르는 cost를 합쳐주고, 모든 S 의 가능성들에 대해서 생각해보면, 최종적으로 이 그래프의 minimum cut은 다음 식이 됩니다.

$$f(L_1, R_1, L_2, R_2) = \min_{S \subset \{L_1, L_1+1, \dots, R_1\}} \left[\sum_{i \notin S} a_i + \sum_{j=L_2}^{R_2} \min \left(\sum_{i \in S} c_i \times d_j, b_j \right) \right]$$

G. 쌀 가져가기와 쿼리

- 최적화를 진행해봅시다. $g(L_1, R_1, x) = S \subset \{L_1, L_1 + 1, \dots, R_1\}$ 를 만족하며 $\sum_{i \in S} c_i = x$ 를 만족하는 모든 S 에 대하여, $\sum_{i \in S} a_i$ 의 최댓값으로 정의합시다. weight가 c_i , value가 a_i 인 knapsack이라고 생각해도 좋습니다. 그러면, 위의 식은 다음과 같이 변형됩니다.

$$f(L_1, R_1, L_2, R_2) = \min_{x=0}^{\infty} \left[\sum_{i=L_1}^{R_1} a_i - g(L_1, R_1, x) + \sum_{j=L_2}^{R_2} \min(x \times d_j, b_j) \right]$$

G. 쌀 가져가기와 쿼리

- b_j 가 20이하라는 것을 이용하면, x 가 20 이상이면 $\sum_{j=L_2}^{R_2} \min(x \times d_j, b_j)$ 값이 $\sum_{j=L_2}^{R_2} b_j$ 로 일정해지는 것을 알 수 있습니다. $\sum_{i=L_1}^{R_1} a_i - g(L_1, R_1, x)$ 는 항상 0 이상이며, $x = \sum_{i=L_1}^{R_1} c_i$ 일때 정확하게 0이 됩니다. 따라서, 식은 다음처럼 적을 수 있습니다.

$$\min \left[\min_{x=0}^{19} \left(\sum_{i=L_1}^{R_1} a_i - g(L_1, R_1, x) + \sum_{j=L_2}^{R_2} \min(x \times d_j, b_j) \right), \sum_{j=L_2}^{R_2} b_j \right]$$

- 결론적으로, 각 쿼리마다 답의 후보는 최대 21개입니다. 21개의 답의 후보들을 전부 계산한 후, 최솟값을 계산하면 됩니다.

G. 쌀 가져가기와 쿼리

- $\sum_{i=L_1}^{R_1} a_i, \sum_{j=L_2}^{R_2} b_j$ 는 point update range sum segment tree를 이용하면 쉽게 계산이 가능합니다. $\sum_{j=L_2}^{R_2} \min(x \times d_j, b_j)$ 역시 각 x 마다 point update range sum segment tree를 만들어서 관리하면 쉽게 계산이 가능합니다. 전부 합쳐서 point update range sum segment tree는 총 22개가 필요합니다.
- $g(L_1, R_1, x)$ 는 knapsack을 monoid의 원소로 가지는 segment tree를 작성하면 쉽게 관리가 가능합니다(구조체 세그먼트 트리라고도 불립니다). 이러한 segment tree는 총 1개가 필요합니다.
- 최종적으로 이 문제는 시간복잡도 $\mathcal{O}((N+Q) \max(b)^2 \log N + (M+Q) \max(b) \log M)$ 에 해결이 가능합니다. $\max(b)$ 는 배열 b 의 최댓값인 20입니다.

H. README

의도된 난이도 **Gold III**

#graphs #bfs

출제자

정우현 (woohyun_jng)

제출 36번, 정답 24팀

정답률: 66.67%

처음 풀 팀

(공식) NJG / 20분

(비공식) intiger / 29분

H. README

- 먼저 모든 정점 v 에 대해 $e(v)$ 를 구해, $e(v) = R$ 인 정점 v 를 하나 찾습니다.
- 각 정점을 시작점으로 BFS를 한 번씩 수행하면 모든 정점까지의 최단 거리를 알 수 있으므로, $e(v)$ 역시 구할 수 있습니다.
- BFS를 모든 정점에서 수행해 $O(NM)$ 의 시간이 필요합니다.
- 또는 플로이드-워셜 알고리즘으로 모든 정점 쌍의 최단 거리를 $O(N^3)$ 에 구해도 제한을 충분히 만족합니다.

H. README

- 이제 $e(v) = R$ 인 정점 v 를 루트로 하여 BFS 트리 T 를 구성합니다.
- BFS 트리에서는 루트 v 에서 모든 정점 u 까지의 거리가 원래 그래프에서의 최단 거리와 같습니다.
- 따라서 T 에서 v 의 이심률은 $e_T(v) = R$ 이므로, T 의 반지름은 R 이하입니다.
- 한편 T 는 G 의 스패닝 트리이므로 간선을 삭제한 그래프입니다. 따라서 임의의 두 정점 사이의 거리는 줄어들 수 없고,

$$\text{dist}_T(x, u) \geq \text{dist}_G(x, u)$$

입니다.

H. README

- 그러므로 모든 정점 x 에 대해 $e_T(x) \geq e_G(x) \geq R$ 이고, T 의 반지름은 R 이상입니다.
- 두 결과를 합치면 T 의 반지름은 정확히 R 입니다. 따라서 v 를 루트로 하는 BFS 트리의 간선 $N - 1$ 개를 출력하면 됩니다.

I. 책 정렬

의도된 난이도 **Diamond IV**

#dp #bitmasking

출제자
pyb1031

제출 57번, 정답 4팀
정답률: 7.02%

처음 푼 팀
(공식) 中國人 / 106분
(비공식) intiger / 155분

I. 책 정렬

- 시행을 총 k 번 했다고 생각해봅시다.
- 각 책마다 길이가 k 인 이진수를 라벨링하는데 i 번째 시행에서 옮겨진 책들의 i 번째비트를 1로 했다고 칩시다.
- 그러면 큰 수가 라벨링된 책은 작은 수가 라벨링된 책의 오른쪽에 있게 됩니다. 같은 수가 라벨링 돼있다면 원본 수열에서의 순서를 따릅니다.
- 즉, 적절히 라벨링하여 k 를 최소화 하며 1비트의 개수를 최소화하는 문제가 됩니다.
- i 번째 책의 위치를 $idx[i]$ 라 합시다. 이 때 $idx[i] < idx[i+1]$ 이라면 $i, i+1$ 번째 책은 같은수를 라벨링하는것이 최적입니다.
- 라벨링해야하는 서로 다른 수의 개수를 x 라하면 k 는 $\lceil \log_2 x \rceil$ 가 됩니다.

I. 책 정렬

- 이제 옮기는 책의 수를 최소화 해야합니다.
- 이는 라벨링된 수들에서 1들의 개수를 최소화 하는것과 같습니다.
- 같은 수를 라벨링해야 하는 책들을 한 그룹으로 묶고 그 그룹의 크기를 w_i 라고 합시다.
- $f(x)$ 를 x 를 이진수로 나타냈을 때 1의 개수라고 정의할 때 $\sum w_i f(s_i)$ 의 합이 최소가 되도록 증가수열 s 를 정하는 문제가 됩니다.
- $dp[i][x]$ 를 i 번째 그룹까지 봤으며 i 번째 그룹에 x 를 라벨링 했을 때 총 1의 개수의 최소값이라고 정의하면 $O(n^2)$ 에 해결할 수 있습니다.

J. 트리 뽑아내기

의도된 난이도 **Platinum III**

#tree #greedy

출제자

김유담 (realpsdoingdamyoo)

제출 28번, 정답 15팀

정답률: 53.57%

처음 푼 팀

(공식) temporary_teamname / 115분
(비공식) intiger / 67분

J. 트리 뽑아내기

- 트리에서 인접한 정점 u, v 에 대해, u 를 제거한 후 v 를 포함하는 컴포넌트의 크기를 sz_{uv} 라고 정의합니다.
- p 와 인접한 정점 $q_1, q_2, \dots, q_m$ 이라고 할 때, $sz_{pq_1}, sz_{pq_2}, \dots, sz_{pq_{\deg(p)}}$ 을 내림차순 정렬한 값들을 $c_{p,1}, c_{p,2}, \dots, c_{p,\deg(p)}$ 이라고 합니다.
- 답은 $A_K = \max_p c_{p,K+1}$ 입니다.

J. 트리 뽑아내기

- 먼저 $A_K \leq \max_p c_{p,K+1}$ 임을 증명합니다.
- 트리의 연결된 두 점 u, v 에 대해 $sz_{uv} \leq \max_p c_{p,K+1}, sz_{vu} \leq \max_p c_{p,K+1}$ 이면 정점 u, v , 간선 uv 를 포함하는 그래프를 H 라고 정의합니다. 만약 그런 두 점이 없다면 H 는 1번 정점만을 포함하는 그래프로 정의합니다.
- H 가 연결 그래프임은 자명합니다.
- 어떤 p 에 대해 p 와 인접하고 H 에 포함될 수 있는 정점은 최대 K 개이므로 그래프는 조건을 만족합니다.
- H 를 제거했을 때 남는 그래프에 크기 $\max_p c_{p,K+1}$ 이상의 컴포넌트가 있다고 가정하면, 그러한 컴포넌트에서 H 의 점과 인접한 x , H 에서 x 와 인접한 y 에 대해 간선 xy 가 포함되지 않았으므로 모순입니다.
- 따라서 $A_K \leq \max_p c_{p,K+1}$ 가 성립합니다.

J. 트리 뽑아내기

- $A_K \geq \max_p c_{p,K+1}$ 임을 증명합니다.
- $\max_p c_{p,K+1} = c_{v,K+1}$ 인 v 인 v 를 잡는다고 할 때, 다음이 성립합니다.
- 최적해의 정점 집합이 v 를 포함하는 경우 : 크기가 각각 $c_{v,1}, c_{v,2}, \dots, c_{v,deg(c)}$ 인 컴포넌트들 중 K 개 이하만이 최적해와 교집합을 갖습니다. 따라서 $A_K \geq c_{v,K+1}$ 입니다.
- 최적해의 정점 집합이 v 를 포함하지 않는 경우 : 크기가 각각 $c_{v,1}, c_{v,2}, \dots, c_{v,deg(c)}$ 인 컴포넌트들 중 1개만이 최적해와 교집합을 갖습니다. 따라서 $A_K \leq c_{v,2} + 1$ 이 되므로, 최적해가 될 수 없습니다.

K. 술 푸는 사람

의도된 난이도 **Platinum I**

#number_theory #constructive

출제자

조완진 (cohenpf)

제출 43번, 정답 20팀

정답률: 53.49%

처음 풀 팀

(공식) 찬우없는방 / 138분
(비공식) 네! 다음은 재밌고 신박한
팀명의 목록입니다. / 160분

K. 술 푸는 사람

- 결국 0부터 시작하여 자연수 a 와 b 를 더하고 빼서 N 이하의 임의의 양의 정수 X 를 최악의 경우 몇 개의 항으로 만들 수 있는냐는 질문과 같습니다. x, y 를 각각 a, b 의 정수 계수라고 하면 다음과 같이 나타낼 수 있습니다.

$$X = x \times a + y \times b$$

이때 국자를 사용한 횟수는 $|x| + |y|$ 입니다.

- 우선 N 이하의 모든 X 를 나타내기 위해서 a 와 b 가 서로소여야 한다는 사실을 알 수 있습니다.
- BFS를 통한 탐색을 떠올릴 수 있으나 (a, b) 쌍에 대해 브루트포스로 접근하면 가능한 쌍이 $O(N^2)$ 개이고 각 쌍에 대해 최대 항의 개수를 구하는 작업은 $O(N)$ 으로, 총 $O(N^3)$ 이 되어 유효한 풀이가 아닙니다.

K. 술 푸는 사람

- 잠시 시선을 돌려 문제에서 요구하는 다른 값, 국자를 사용해야 하는 횟수를 살펴봅시다. 국자를 최대 k 번 사용한다고 하겠습니다. 문제에서 요구하는 것은 주어진 N 에 대한 k 의 하한입니다. 이를 어떻게 구할 수 있을까요?
- 관점을 뒤집어 봅시다. $|x| + |y| \leq k$ 일 때, $X = x \times a + y \times b$ 로 나타낼 수 있는 정수 X 는 모두 몇 개일까요?
- 서로 다른 순서쌍 (x, y) 에 대해 X 의 값이 겹치지 않는 이상적인 경우, 표현할 수 있는 X 의 개수는 곧 $|x| + |y| \leq k$ 인 영역 안의 정수점 개수와 같습니다. 이 영역은 2차원 좌표평면에서 마름모, 즉 45° 회전한 정사각형 모양입니다. 이 안의 정수점 개수는 다음과 같습니다.

$$2k^2 + 2k + 1$$

K. 술 푸는 사람

- $|x| + |y| \leq k$ 를 만족하는 순서쌍 (x, y) 은 $2k^2 + 2k + 1$ 개이고, 각 순서쌍은 하나의 정수 $X = x \times a + y \times b$ 에 대응합니다. 따라서 서로 다른 X 의 개수는 최대 $2k^2 + 2k + 1$ 개입니다.
- 가능한 X 중에는 0이 있고, (x, y) 로 X 를 표현할 수 있다면 $(-x, -y)$ 로 $-X$ 를 표현할 수 있습니다. 따라서 0이 아닌 값은 양수와 음수가 대칭으로 나타납니다.
- 따라서 표현할 수 있는 서로 다른 양의 정수 X 의 개수의 상한은 다음과 같습니다.

$$\frac{2k^2 + 2k + 1 - 1}{2} = k^2 + k$$

K. 술 푸는 사람

- 그런데 문제에서는 1부터 N 까지의 모든 양의 정수, 총 N 개를 표현할 수 있어야 한다는 조건이 주어졌습니다. 앞서 국자를 최대 k 번 사용해 표현할 수 있는 서로 다른 양의 정수의 개수가 최대 $k^2 + k$ 개임을 보였으므로, 다음 부등식이 성립해야 합니다.

$$k^2 + k \geq N$$

- 이로서 문제에서 요구하는 국자 사용 횟수의 하한을 구했습니다.
- 이제 $k^2 + k \geq N$ 을 만족하는 가장 작은 양의 정수 k 에 대해, 실제로 국자를 최대 k 번 사용하여 N 이하의 모든 양의 정수를 만들어 낼 수 있다면 문제가 해결됩니다.
- 이를 가능하게 하는 a 와 b 가 항상 존재할까요? 답은 그렇다입니다.

K. 술 푸는 사람

- a 와 b 를 다음과 같이 정합니다.

$$a = k, \quad b = k + 1$$

- $1 \leq i \leq k$ 인 i 에 대해, a 와 b 를 합쳐 정확히 i 번 더해서 만들 수 있는 정수는 다음 구간을 이룹니다.

$$[ik, i(k+1)]$$

- 연속한 두 구간 사이에는 $((i-1)(k+1), ik)$ 에 속하는 $k-i$ 개의 정수가 남습니다. 왼쪽 끝에서 d 만큼 떨어진 정수는 다음과 같이 표현할 수 있습니다.

$$(i-1)(k+1) + d = -dk + (i-1+d)(k+1)$$

이때 국자 사용 횟수는 $i-1+2d$ 입니다.

K. 술 푸는 사람

- 오른쪽 끝에서 d 만큼 떨어진 정수도 다음과 같이 표현할 수 있습니다.

$$ik - d = (i + d)k - d(k + 1)$$

이때 국자 사용 횟수는 $i + 2d$ 입니다.

- 왼쪽에서 $\left\lceil \frac{k-i}{2} \right\rceil$ 개, 오른쪽에서 $\left\lfloor \frac{k-i}{2} \right\rfloor$ 개를 채우면 두 경우 모두 국자 사용 횟수가 k 이하입니다. 따라서 두 구간 사이의 $k-i$ 개 정수를 모두 표현할 수 있습니다.
- 그러므로 다음과 같이 정한 k, a, b 가 문제의 답 중 하나가 됩니다.

$$k = \min\{t \in \mathbb{Z}_{>0} \mid t^2 + t \geq N\}, \quad a = k, \quad b = k + 1$$

실제로 국자를 최대 k 번 사용하여 1부터 $k^2 + k$ 까지의 모든 정수를 만들 수 있고, 이는 앞서 구한 하한과 일치합니다.

K. 술 푸는 사람

- 이 문제는 작년(2025년)까지는 OpenAI 및 Gemini의 공개된 LLM 모델들이 풀지 못했습니다.
- 아쉽게도 대회가 열린 시점에는 아니게 되었습니다.
- 저는 출제자라서 증명을 해야했지만 증명 없이 문제만 풀자면 작은 크기에서 브루트포스를 해보고 규칙 찾기를 하는게 훨씬 쉽습니다.

L. 정투상도

의도된 난이도 **Gold III**

#implementation #bruteforcing

출제자

조완진 (cohenpf)

제출 59번, 정답 23팀

정답률: 40.68%

처음 풀 팀

(공식) 中國人 / 44분

(비공식) intiger / 23분

L. 정투상도

- 각 (x, y, z) 위치에 블록을 놓을 수 있는지 생각해 봅시다.
- 해당 위치에 블록이 놓여 있다면 정면도, 평면도, 우측면도에서 그 위치에 대응하는 칸이 모두 블록이 보여야 합니다. 따라서 세 도면의 대응하는 칸이 모두 표시된 경우에만 (x, y, z) 에 블록을 놓을 수 있습니다.
- $N \times M \times K \leq 10^6$ 이므로 모든 격자 칸을 직접 확인할 수 있습니다. 가능한 모든 위치에 블록을 놓아 최대 블록 배치를 구성합니다.
- 올바른 정투상도가 주어졌다면 이 배치의 정투상도는 입력으로 주어진 도면과 일치합니다. 아니라면 Wrong design을 출력하고 여기서 끝내면 됩니다.

L. 정투상도

- 이제 최대 블록 배치에서 각 투영선에 놓인 블록의 개수를 셉니다. 즉 정면도, 평면도, 우측면도의 각 칸에 대응하는 직선마다 블록이 몇 개 있는지 기록합니다.
- 어떤 위치 (x, y, z) 의 블록을 제거해도 정투상도가 변하지 않으려면, 이 블록이 속한 세 투영선 모두에 다른 블록이 남아 있어야 합니다.
- 따라서 세 투영선에 놓인 블록의 개수가 모두 2 이상이라면 해당 블록은 제거할 수 있습니다. 하나라도 1이라면 그 블록을 제거했을 때 해당 도면의 칸이 사라지므로 제거할 수 없습니다.

L. 정투상도

- 각 (x, y, z) 마다 대응하는 세 투영선 위의 블록 개수를 확인합니다.
 - 투영선 상의 블록 개수가 모두 2 이상이면 1을 출력합니다.
 - 그렇지 않으면 2를 출력합니다.
- 최대 블록 배치를 구성하면서 각 투영선의 블록 개수를 함께 세면, 이후 각 위치의 답을 상수 시간에 결정할 수 있습니다.
- 모든 격자 칸을 상수 번씩 확인하므로 전체 시간복잡도는 $O(NMK)$ 입니다.
- 사실 원본 문제는 BOJ 2009 Minecraft와 notorious coincidence였던 사실이 밝혀져서 형식이 지금과 같이 변경되었습니다.
 - 해당 문제의 난이도는 Silver II였습니다.
 - ★ 그러니까 이 문제도 Silver II라고 주장을 하고 싶었는데 검수진분들이 아니래요.

M. 쌀 재배하기와 퀴리

의도된 난이도 **Platinum III**

#ad_hoc #prefix sum

출제자

임병찬 (hyperbolic)

제출 76번, 정답 13팀

정답률: 17.11%

처음 푼 팀

(공식) temporary_teamname / 94분

(비공식) intiger / 111분

M. 쌀 재배하기와 쿼리

- 각 칸 (i, j) 마다, $U = D = i, L = R = j$ 로 초기화를 해줍니다.
- 그후, $N + M$ 번 다음과 같은 행동을 순서대로 반복합니다.
 - 오른쪽으로 더 확장해야 한다면 오른쪽으로 확장합니다. 구체적으로, 만약 $U \leq k \leq D$ 를 만족하고 $a_{k,R} = a_{k,R+1}$ 을 만족하는 k 가 존재한다면, R 을 1만큼 늘립니다.
 - 왼쪽으로 더 확장해야 한다면 왼쪽으로 확장합니다. 구체적으로, 만약 $U \leq k \leq D$ 를 만족하고 $a_{k,L} = a_{k,L-1}$ 을 만족하는 k 가 존재한다면, L 을 1만큼 줄입니다.
 - 위쪽, 아래쪽에 대해서도 마찬가지로 행동합니다.
- 이 행동을 반복한 후, (U, D, L, R) 이 (i, j) 좌표에 대한 답이 됩니다. 시간은 총 $\mathcal{O}(N + M)$ 이 소요되므로, 모든 좌표에 대해 답을 구하는데 총 $\mathcal{O}(NM(N + M))$ 의 시간이 걸립니다.

M. 쌀 재배하기와 쿼리

- 이 방법이 답이 된다는 것을 어떻게 증명할 수 있을까요?
- (i, j) 칸에 대한 답이 되는 직사각형을 X 이라고 정의합시다. 또한, 위 방법으로 얻어낸 직사각형을 Y 라고 정의합시다.
- X 와 Y 는 모두 조건을 만족하는 직사각형입니다. 이 경우, $X \cap Y$ 역시 조건을 만족하는 직사각형인것을 알 수 있습니다. 그러나 X 는 조건상 가장 작은 크기가 되어야 하기에 $X \cap Y = X$ 여야 합니다. 이는 $X \subset Y$ 를 의미합니다.
- 또한 Y 의 경우, 알고리즘의 동작 방법상 X 를 벗어날 수 있는 방법이 없습니다. 그러므로, $Y \subset X$ 가 성립합니다. 최종적으로, $X = Y$ 가 성립합니다.

M. 쌀 재배하기와 쿼리

- 알고리즘에서 확장 가능성은 2차원 누적합 배열을 이용하면 $\mathcal{O}(1)$ 에 가능합니다. 오른쪽을 예시로 들면, 다음과 같이 판단이 가능합니다.
- $\text{right}[i][j] = (a[i][j] = a[i][j+1])$ 이 성립하면 1, 아니면 0) 으로 정의합시다. 오른쪽으로 확장해야 한다는 것은 $\sum_{i=U}^D \text{right}[i][R] > 0$ 인 것과 동치가 됩니다.

N. 매직 쌀 2

의도된 난이도 -

#constructive #case work #implementation #precomputation

출제자

윤지호 (annyeong1)

제출 7번, 정답 0팀

정답률: 0.00%

처음 푼 팀

(공식) 없음 / -

(비공식) 없음 / -

N. 매직 쌀 2

- $N \times N$ 격자의 맨 왼쪽 위에만 칠해져 있고, 이미 칠해져 있는 칸을 ㄱ자로 둘러싸는 세 칸을 칠하는 것을 반복한다고 생각하면 좋습니다.
- Fun Fact: 빈칸 없이 칠할 수 있음과 N 이 2의 거듭제곱꼴임이 필요충분입니다.
- 재귀를 이용하는 ㄱ자 타일링을 생각하면 한 방향은 쉽고, 역방향은 많이많이 어렵습니다.
- 궁금하신 분들은 2024 IMO Shortlist C8을 참고하시면 되겠습니다.
- 아쉽게도 이 문제에서 2의 거듭제곱으로 시작하면 문제를 풀기 어렵습니다.

N. 매직 쌀 2

ㄱ자를 그려보면서 다음을 관찰합시다:

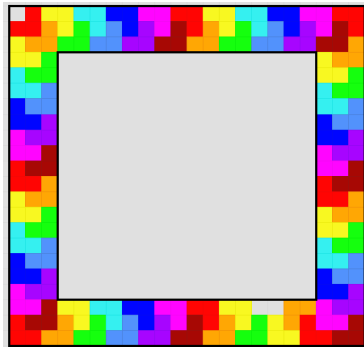
- 한 칸이 튀어나와 있어야 이를 둘러싸는 세 칸 기억자를 그릴 수 있습니다.
- $3 \times N$ 의 긴 '막대'를 뽑을 수 있을 것 같아 보입니다.
- 긴 막대는 길이를 2씩 늘릴 수 있어 보이고, 중간에 꺾을 수 있습니다.
- 이 막대로 잘 평면을 채우면 빈칸을 10개 안쪽으로 줄일 수 있을까요?

N. 매직 쌀 2

- 막대를 잘 구부려서 k 자로 정사각형을 채우는 것을 시도합니다. 가로세로가 3의 배수이면 적당히 채울 수 있을 것 같은 느낌입니다.
- 그런데 3의 배수가 아니면 $O(N)$ 개 정도의 빈칸이 남는 것 같습니다. 사실 N 이 홀수일 때도 적당히 채우는 것이 많이 어렵습니다.
- 앞에서 막대 길이가 2씩 늘어난다고 생각하는 쪽이 편함을 관찰했으니, 일단 N 이 짝수일 때만 생각합니다.
- 한 쪽부터 k 자로 채우는 건 별로 건강에 안 좋아 보입니다. 약간 두께가 3인 테두리를 연속적으로 감아서 작게 만든 다음 작은 케이스를 노가다하는 생각을 할 수 있을까요?

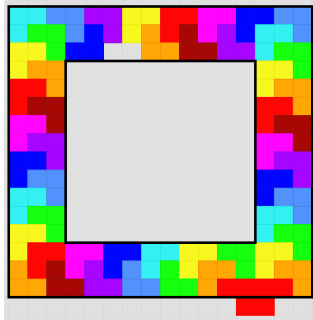
N. 매직 쌀 2

- 이렇게 하면 적당히 두께가 3인 테두리를 만들 수 있습니다. 테두리의 오른쪽 아래 두 칸이 비어있음을 확인합시다.



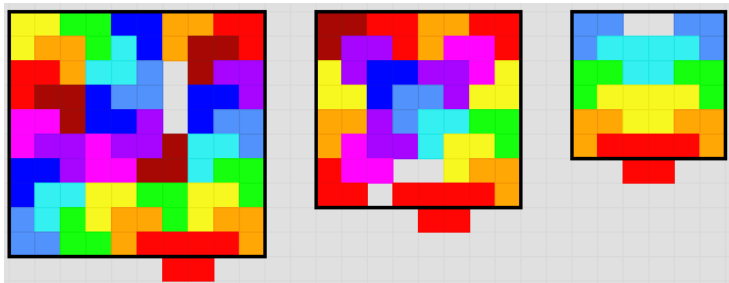
N. 매직 쌀 2

- 오른쪽 아래 빈칸 두 개에서 시작해서 또 채워봅시다. 똑같이 오른쪽 아래에 빈칸이 두 개 남으면 재귀적으로 풀 수 있으니 좋겠습니다.
- 해보면 잘 안 됩니다. 대신 왼쪽 위에 빈칸을 두 개 남기는 건 가능합니다.



N. 매직 쌀 2

- 그럼 점대칭적으로 잘 줄일 수 있겠습니다. N 이 충분히 작아질 때까지 한 다음 작은 케이스는 전처리하면 되겠습니다.
- 정해는 N 이 6, 8, 10일 때를 전처리합니다.

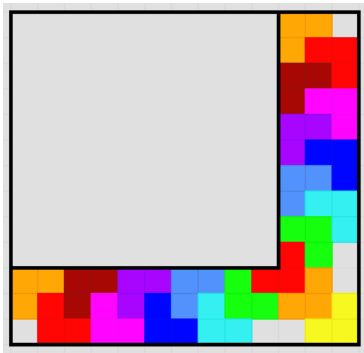


N. 매직 쌀 2

- 이렇게 하면 N 이 짝수일 때 빈칸을 3개 이하만 쓰고 타일링을 할 수 있습니다. 이제 N 이 홀수일 때를 처리해주어야 합니다.
- $N-1$ 을 만들어두고 가로세로 한 줄을 추가해주는 건 말이 안 되는 것 같습니다.
- $N-3$ 을 만들어두고 가로세로 세 줄을 추가해주는 건, 앞에서 했던 두께 3의 막대를 잘 그리면 비슷한 방법으로 할 수 있지 않을까요?

N. 매직 쌀 2

- 아래처럼 하면 됩니다:



N. 매직 쌀 2

- 이렇게 하면 빈칸이 6개 추가됩니다. 따라서 빈칸을 9개 안쪽으로 쓰고 타일링을 하는 것이 가능합니다.
- 저는 풀이를 만드는 데 5시간 내외, 구현하는 데 1시간 정도가 걸렸습니다.
- 대회 5일 전에 (참가자 목록과 그 실력대를 보고) 급하게 출제해서, 남은 기간동안 솔루션 작성에 성공한 검수진이 없었습니다. 따라서 예상 난이도가 없습니다.
- 저는 P3..D4 정도 주면 적당할 것이라 생각했습니다만, 별개로 구현이 더럽게 오래 걸리므로 올솔 방지에는 탁월할 것이라 생각해 출제했습니다. 유일하게 안 풀린 문제이므로 목표는 다한 것 같습니다.