

Coder's High 2013 Solution

The Problemsetters

July 28, 2013

A : xhaeneung (출제: LIBe)

문자열 C 를 오름차순으로 정렬해서 푸는 게 좋고, 영어 숫자에 대해서 사전을 만들면 좋음.

B : Four gods (출제: Kureyo, LIBe, xhae)

- u, v 를 잡고 u, v 에 인접한 점들을 찾고, $\binom{n}{2}$.
- 중복 제거를 위해 인접한 애들 검색할 때 후보를 ???

C : Jump! Jump! (출제: Astein)

$$\left(\frac{n(n+1)}{2} + 1 \right)^2 \pmod{20130728}$$

D : Space (출제: altertain)

0 - 255 까지 보는 대신,

- $S_x = \{0, 255, x_{\min}^{(1)} - 1, \dots, x_{\min}^{(20)} - 1, x_{\max}^{(1)} + 1, \dots, x_{\max}^{(20)} + 1\}$
- $S_y = \{0, 255, y_{\min}^{(1)} - 1, \dots, y_{\min}^{(20)} - 1, y_{\max}^{(1)} + 1, \dots, y_{\max}^{(20)} + 1\}$
- $S_z = \{0, 255, z_{\min}^{(1)} - 1, \dots, z_{\min}^{(20)} - 1, z_{\max}^{(1)} + 1, \dots, z_{\max}^{(20)} + 1\}$

로 줄여서 (각각 42개씩) 체크하면 됩니다.

E : Travel in xhaeland (출제: Astein)

금지구역 기준 8방향의 모든 점까지 BFS로 계산.

- 계산된 점끼리 연결할 수 있는지 (사각형 내에 금지구역 없어야)
- 계산된 점에서 직선을 그렸을 때 다른 점으로 갈 수 있는지.

F : 단짝 문자열 (출제: altertain)

- $prev(i, j) = \sim i, \sim j$ 까지 공통 문자열 길이
- $next(i, j) = i\sim, j\sim$ 까지 공통 문자열 길이

각 i, j 에 대해 $\min(prev(i, j), next(i, j)) \implies L$

G : Railgun (출제: xhae)

500원이 Acc_i 에 있으며, 이전에 온 위치가 Acc_j 일 때, (i, j) pair를 확정하면 이 다음에 갈 수 있는 (x, i) pair들을 확정할 수 있다.

따라서 50×50 개의 node를 BFS 한다.

H : Weekly Calendar (출제: Kureyo)

달력에 대한 깊은 이해 + 오타 주의 ^^

I : The hundred years' war (출제: Kureyo)

- pre-order로 나열하면 subtree 는 구간 문제로 바꿀 수 있다.
- 반란을 구간 XOR 1 문제로 바꾸면 indexed-tree 문제로 풀 수 있음.

J : Book Thief (출제: Kureyo)

$$\begin{aligned} DP(N, V) &= \max \begin{cases} DP(N - 1, V) \\ DP(N - 1, V - v_i) + C_i \\ \vdots \\ DP(N - 1, V - v_i \cdot t) + C_i \cdot t \end{cases} \\ &= \max \left\{ DP(N - 1, b) + \left\lfloor \frac{V - b}{v_i} \right\rfloor \cdot C_i \right\} \\ &= \max \left\{ DP(N - 1, b) - \left\lfloor \frac{b}{v_i} \right\rfloor \cdot C_i + \left\lfloor \frac{V}{v_i} \right\rfloor \cdot C_i \right\} \end{aligned}$$

where $V - v_i \cdot k_i \leq b \leq V$, $b \equiv V \pmod{v_i}$

J (cont'd)

$$DP(N, V) = \max \left\{ DP(N-1, b) - \left\lfloor \frac{b}{v_i} \right\rfloor \cdot C_i + \left\lfloor \frac{V}{v_i} \right\rfloor \cdot C_i \right\}$$

- priority queue 사용 : $O(NV \log V)$
- $V_1 < V_2$ 이고, $D(N, v_1) - \left\lfloor \frac{v_1}{v_i} \cdot C_i \right\rfloor > D(N, v_2) - \left\lfloor \frac{v_2}{v_i} \cdot C_i \right\rfloor$ 이면 v_2 는 가능한 max 값이 될 수 없다.
- 따라서, stack 형식으로 만들면 amortized linear $O(NV)$ 에 풀 수 있음.

K : Clean - 어떤 과학의 인공지능 청소기 (출제: kcm1700)

N, M 이 작아서 모든 경우를 다 돌려도 순식간에 답을 찾을 수 있다.

L : k번째 사전순 숫자 (출제: Kureyo)

DP (현재 자리수, lower bound를 이미 넘어섰는지 여부, upper bound 미만이 보장되는지 여부).