

G: エレベーター

原案 : vvataarne

解説 : vvataarne

問題概要

- 2つのクエリを処理する
- エレベータ設置クエリ
 - A_i 階以上 B_i 階以下を行き来可能にする
- 移動可能か判定クエリ
 - S_i 階から T_i 階に移動可能か

考察

- 時系列順に処理することを考える
- 昼と夜の順を間違えないように気を付ける

想定TLE

- 階を頂点とみなしてグラフを作る
- 初期状態として i 階から $i - 1$ 階に辺を張る
- エレベータ設置クエリ $O(1)$
 - A_i 階から B_i 階に辺を張る
- 移動可能か判定クエリ $O(M + N)$
 - S_i 階から T_i 階に到達可能かBFSで判定

考察

- 階段を使うと下の階にはいつでも行ける
 - 以降 $s_i < T_i$ のみを考える

想定TLE その2

- UnionFindを使う
 - エレベータ設置クエリ $O(N\alpha(N))$
 - A_i 階が属する集合と A_{i+1} 階が属する集合を合併
 - A_{i+1} 階が属する集合と A_{i+2} 階が属する集合を合併
 - ...
 - B_{i-1} 階が属する集合と B_i 階が属する集合を合併
 - 移動可能か判定クエリ $O(\alpha(N))$
 - S_i 階が属する集合と T_i 階が属する集合が等しいかで判定
- (α はアッカーマン関数の逆関数)

考察

- 想定TLEその2の高速化を考える
- エレベータ設置クエリ $O(N\alpha(N))$
 - 区間 unite を高速に処理できるデータ構造が欲しい

想定解

- UnionFindを使う
- RangeUpdateQuery対応のデータ構造(セグメント木)を使う
- エレベータ設置クエリ $O(\log(N))$
 - A_i, \dots, B_i 階が属する集合を S_i 階が属する集合で更新
- 移動可能か判定クエリ $O(\log(N))$
 - S_i 階が属する集合と T_i 階が属する集合が等しいかで判定

総評

- AC/Submit …… 21/95 (22.11%)
- First AC
 - オンサイト …… rupc_ei13 (38:33)
 - オンライン …… rupc_ei13 (38:33)