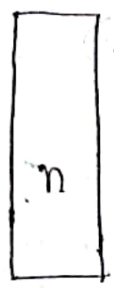
Game theory

- → Two or more player
- → sequential moves
- → partial game / impartial game.
- → state :
- → Winning / losing state.

problem

① Mirror move (problem).

② pattern



pick ≤ 2

Only one pile with n stones

n	first
0	L
1	W
2	W
3	L
4	W
5	W
6	L
7	W
⋮	⋮

যখন K টা পাথর নিতে পারবে তাহলে first player এর জন্য lossing state হবে।

if $\{n \% (k+1) == 0\}$ তাহলে loss হবে

otherwise win হবে।

W node = If has an edges to a losing node

L node = All edges result to a winning node.

a_1 a_2 . . . a_n

piles.

Let a_k be the first non-empty pile. then the player choose some stones from a_k.

Who cannot pick any stone loose the game.

⇒ problem link (sequential move) CF

NIM Theory:

a_1 a_2 . . . a_n

প্রতি প্লেয়ার যে কোন পাইল হতে কিছু সংখ্যক Stone সরাবে।

যে কোনো stone remove করতে পারবে না সে হারবে।

$x = a_1 \oplus a_2 \oplus \cdots \oplus a_n$

If

xor > 0	Winning state for first player
xor = 0	Lossing state for first player

Type 2:

যেকোন nonempty piles হতে $\le K$ সংখ্যক stones সরানো যাবে। সর্বশেষ ~~যে~~ stone যে সরাবে সেই ব্যক্তি জিতবে।

Soln:

$$xor = \{a_1 \% (K+1)\} \oplus \{a_2 \% (K+1)\} \oplus \cdots \oplus \{a_n \% (K+1)\}.$$

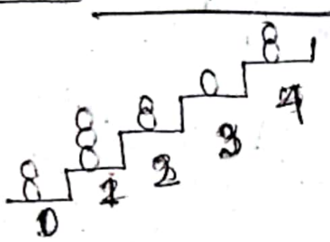
if $xor > 0$ (First player WIN)

else $xor == 0$ (second player WIN).

| problem Link
| CSES → NIM GAME II

Type 3: Staircase NIM

>0



যেকোন staircase(i) হতে যে কোন সংখ্যক stone সরিয়ে (i-1) এ আনতে পারবে। সর্বশেষ যে ব্যক্তি এই কাজটি করতে পারবে সেই জিতবে।

Case: even index কোন কাজে দিবে না কারণ তা সবসময় 1 ইনডেক্স নিচে যায়।

① (>0) odd index এর সব ভ্যালুর XOR যদি (>0) হয় তবে First player WIN.

② অন্যথায় second player WIN.

| problem link
| CSES : Staircase game

problem link: Atcoder dp contest (L - Deque)

Minmax algorithms

একটি Array দেওয়া থাকবে, ২জন প্লেয়ার optimally move করে শুধু Array এর সর্বপ্রথম বা সর্বশেষ এলিমেন্ট নিবে। প্লেয়ার দুইজনেই optimally খেলবে যার অর্থ এলিমেন্ট বাছবে নিবেন; আমাদের জানতে হবে ১ম প্লেয়ার এর মোট স্কোর থেকে ২য় প্লেয়ার এর স্কোর বাদে total_sum of array element — ১ম প্লেয়ার এর অবশিষ্ট স্কোর।

Recursion এ: dp[L][R][Turn].

base case:

যদি ১ম প্লেয়ার এর Turn হয় তবে

→ (L==R) হলে তখন a[L] return করবে

অন্যথায় স্কোর 0 return হবে কারণ ২য় প্লেয়ার নিচ্ছে নিবে তাই ১ম প্লেয়ার এর স্কোর বাড়বে না।

Transition:

১ম প্লেয়ার এর টার্ন:

return হবে max(a[l]+rec(l+1, r, 2), a[r]+rec(l, r-1, 2))

↑ হবে কারণ ১ম প্লেয়ার বেশি স্কোর করতে চাইবে

২য় প্লেয়ার এর টার্ন

return হবে min(rec(l+1, r, 1), rec(l, r-1, 1))

↑ হবে কারণ ২য় প্লেয়ার ১ম প্লেয়ার এর স্কোর কমাতে চাইবে