

Working of Nim-sum Formula

Working of Nim-Sum Formula

- * The game alternates between losing (0 nim-sum) and winning (non-zero nim-sum) states.
- * In losing state, all paths lead to game loss.
- * In winning state, there exists at-least one path that leads to win.

$$\text{nim-sum} = \text{nim}[0] \wedge \text{nim}[1] \wedge \dots \wedge \text{nim}[n-1]$$

1) If nim-sum is 0, then a move makes it non-zero.

Nim Sum Before $\text{nim-sum}_b = \text{nim}[0] \wedge \text{nim}[1] \wedge \dots \wedge \text{nim}[n-1]$
 Nim Sum After $\text{nim-sum}_a = \text{nim}[0] \wedge \text{nim}[1] \wedge \dots \wedge \text{nim}[n-1]$

Note: In these two sequences only one value is different

$$\begin{aligned} \text{nim-sum}_a &= (\text{nim-sum}_b \wedge \text{nim-sum}_b) \wedge \text{nim-sum}_a \\ &= (\text{nim}[0] \wedge \dots \wedge \text{nim}[n-1]) \wedge (\text{nim}[0] \wedge \dots \wedge \text{nim}[n-1]) \wedge (\text{nim}[0] \wedge \dots \wedge \text{nim}[n-1]) \\ &= 0 \wedge (\text{nim}[0] \wedge \text{nim}'[0]) \wedge \dots \wedge (\text{nim}[n-1] \wedge \text{nim}'[n-1]) \\ &= \text{nim}[k] \wedge \text{nim}'[k] \quad \text{Here nim}[k] \text{ is the heap from where the player picked coin(s)} \\ &> 0 \quad [\text{Since } \text{nim}[k] \neq \text{nim}'[k]] \end{aligned}$$

Therefore the game always moves from losing to winning state irrespective of the move

2) There always exists at least one move to change winning state (non-zero nim-sum) to losing state (0 nim-sum)

Note: $\text{nim-sum}_b > 0$ and we need to make nim-sum_a as 0

$$\text{nim-sum}_a = \text{nim-sum}_b \wedge \text{nim}[k] \wedge \text{nim}'[k] \quad \left[\begin{array}{l} \text{We have shown in} \\ \text{proof 1} \end{array} \right]$$

We pick a heap $\text{nim}[k]$ with the most significant bit same as nim-sum_b and change it to $\text{nim-sum}_b \wedge \text{nim}[k]$.
 So putting this value in place of $\text{nim}'[k]$, we get

$$\begin{aligned} \text{nim-sum}_a &= \text{nim-sum}_b \wedge \text{nim}[k] \wedge (\text{nim}[k] \wedge \text{nim-sum}_b) \\ &= 0 \end{aligned}$$

Therefore there always exists a change in $\text{nim}[]$ that changes nim-sum from non-zero to zero.

$\text{nim}[] = \{9, 4, 1\}$
 $\text{nim-sum}_b = 9 \wedge 4 \wedge 1$
 $(1001 \wedge 0100 \wedge 0001)$
 $= 12$
 The number in $\text{nim}[]$ with same MSB as 12 is 9. We replace 9 with $9 \wedge 12 (1001 \wedge 1100)$ which is 5 (101)
 $\text{nim-sum}_a = 5 \wedge 4 \wedge 1$
 $= (101 \wedge 100 \wedge 1)$
 $= 0$

Working of Sprague Grundy Theorem

Working of Grundy Number Theorem

Grundy Number for a Composite Game:

$$g(x_1, x_2 \dots x_n) = g_1(x_1) \wedge g_2(x_2) \dots g_n(x_n)$$

Remember: Grundy Number of a Single Impartial Game

$$g(x) = \text{mex}(\{g\text{-values of all possible states after a move}\})$$

For example, in a single pile game

$$g(x) = \text{mex}(\{g(x - \text{pick}(i)), g(x - \text{pick}(j)), \dots g(x - \text{pick}(x_i - 1))\})$$

Working of Grundy Number Theorem

Grundy Number for a Composite Game:

$$g(x_1, x_2 \dots x_n) = g_1(x_1) \wedge g_2(x_2) \dots g_n(x_n)$$

$$\text{Let } b = g_1(x_1) \wedge g_2(x_2) \dots \wedge g_n(x_n)$$

1) For every non-negative integer $a < b$, the value 'a' must exist among combined g-values after a move.

$$g(x_1, x_2 \dots x_n) = \text{mex}(\{g(x'_1, x_2 \dots x_n), g(x_1, x'_2 \dots x_n), \dots g(x_1, x_2 \dots x'_n)\})$$

2) The value b should not exist after a move.

1) For every non-negative 'a' such that $a < b$, the value 'a' must exist among all possible g-values of the combined game.

$$b = g_1(x_1) \wedge g_2(x_2) \dots g_n(x_n)$$

Proof: There always exists a $g_k(x_k)$ such that if we replace it with the following $g_k(x'_k)$ we get the value 'a'.

$$g_k(x'_k) = g_k(x_k) \wedge a \wedge b$$

The g-value of the combined game after the above move

$$\begin{aligned} &= g_1(x_1) \wedge g_2(x_2) \dots g_k(x'_k) \dots g_n(x_n) \\ &= g_1(x_1) \wedge g_2(x_2) \dots g_k(x_k) \wedge a \wedge b \dots g_n(x_n) \\ &= b \wedge a \wedge b \\ &= a \end{aligned}$$

How to find $g_k(x_k)$?

The leading bit in $a \wedge b$ must be set in b because $b > a$. Therefore there must exist a $g_k(x_k)$ with the same leading set bit. And we can reduce it to $g_k(x_k) \wedge a \wedge b$ because $g_k(x_k) \wedge a \wedge b < g_k(x_k)$.

2) The value 'b' can not exist after a move

Let a move be $g_k(x_k)$ to $g_k(x'_k)$

For the value 'b' to exist after a move

$$g_1(x_1) \wedge \dots g_k(x_k) \wedge g_n(x_n) = g_1(x_1) \wedge \dots g_k(x'_k) \wedge g_n(x_n)$$

which means

$$g_k(x_k) = g_k(x'_k)$$

which contradicts because g-value of a game cannot be same after a move.