Números NIM-Parte I

Vamos falar um pouco sobre o jogo NIM e algumas aplicações, além de algumas variações do jogo. Separaremos o estudo dos Números NIM em 3 partes. Vejamos o objetivo de cada Parte:

1. **Parte I**:Introduzir os conceitos iniciais de números NIM, com ênfase na aplicação, sem muita base matemática, objetivando ver o mais rápido possível algumas aplicações
2. **Parte II:**Fortalecer a base matemática para prosseguir os estudos sobre NIM e ver algumas variações mais complexas do jogo
3. **Parte III:**Aplicar as ideias aprendidas nas partes anteriores para resolver problemas computacionais e matemáticos sobre números NIM.

- **O jogo NIM:** O jogo NIM é jogado por dois jogadores e consiste basicamente de n montes de pedras, onde cada monte possui $a_1, a_2, a_3, \ldots, a_n$ pedras. Cada jogador pode, em sua rodada, escolher um monte de pedras e retirar quantas pedras desejar. Quem retira a última pedra ganha o jogo. Você também pode ler esse:link.
- **Operador Xor:** O operador XOR será representado por "^". Ele satisfaz:
  F^F=F
  V^F=V
  F^V=V
  F^F=0
  Ou seja, ele assume valor verdade, se as duas variáveis forem distintas.
  Você pode encontrar um pouco mais sobre o operador XOR aqui.
  Além disso, você pode resolver o seguinte problema neste link.
  Vamos aplicar o operador XOR em alguns exemplos:
  Exemplo 1: Calcular 5^3
  Solução:O primeiro passo é converter os números para a base 2:
  5 = $(101)_2$ e 3 = $(11)_2$.
  Aplicando o XOR, teremos:

```
    1 0 1
^     1 1
 ----------
    1 1 0
```

  Veja que iniciamos da direita pra esquerda, 1^1 = 0, depois 0^1 = 1 e 1 = 1.
  Exemplo 2: Calcular 7^4
  Solução: 7 = $(111)_2$ e 4 = $(100)_2$
  Aplicando o XOR, teremos:

```
    1 1 1
^   1 0 0
 ----------
    0 1 1
```

  Ou seja, quando for fazer Xor com inteiros, você deve sempre transformar os inteiros pra base 2, depois aplicar o xor em cada coluna.

- **A estratégia vencedora:** Uma configuração ($a_1, a_2, a_3, \ldots, a_n$) , ou seja, n montes com $a_1, a_2, a_3, \ldots, a_n$ pedras é perdedora, se e somente se, $a_1 \wedge a_2 \wedge a_3 \wedge \ldots \wedge a_n$ é zero.
  Você pode encontrar uma prova para isso, além de muito mais informações sobre teoria dos

jogos no seguinte link.

- **Uma explicação informal:**
    1. Note que de posições perdedoras (XOR total da configuração é zero), ao modificarmos qualquer monte de pedras, iremos fazer o XOR total virar diferente de zero, portanto "entregaremos" uma posição vencedora.
    2. Note que de posições vencedoras sempre é possível "entregar" uma perdedora, pois imagine que estamos numa posição vencedora. Logo, o XOR total não é zero. Portanto, existe alguma coluna que possui um número 1 como resposta. Tome a coluna mais à esquerda. Daí, mude o valor de algum 1 dessa coluna para 0, e então podemos ir "arrumando" todos os números à direita dessa coluna, trocando 1 por 0, ou 0 por 1 quando necessário...A ideia é que $2^n > 2^0 + 2^1 + \ldots + 2^{n-1}$. (Exercício:Entender a explicação, ou procurar na internet e entender o por que a estratégia funciona)
- **Misere NIM:** Misere NIM é uma pequena variação do NIM tradicional. A grande diferença é que no Misere NIM, o último jogador que retira pedra é o perdedor.
- **Estratégia:** Se o início do jogo é formado por vários montes com exatamente 1 pedra, teremos que se houver um número par de montes, o jogador 1 é quem tem a estratégia vencedora, e se houver um número ímpar de montes, o jogador 2 é quem tem a estratégia vencedora. Se no início houver algum monte com mais de uma pedra, teremos que uma configuração ($a_1, a_2, a_3, \ldots, a_n$) é perdedora, se e somente se, $a_1 \wedge a_2 \wedge a_3 \wedge \ldots \wedge a_n$ é zero. (Exercício:Provar que essa estratégia funciona, ou encontrar e entender alguma solução na internet...OBS:Existe a solução no seguinte link.)

Aplicações:

Problema 1. (NIM) Crie um programa que leia um inteiro k. Depois leia k inteiros $a_1, a_2, a_3, \ldots, a_k$ e diga se quem tem a estratégia vencedora é o primeiro ou o segundo jogador.
Link para o problema:
http://acm.mipt.ru/judge/problems.pl?problem=100&lang=en
Problema 2. (Jogos)
http://www.codechef.com/problems/RESN04#
Problema 3.(MISERE NIM)
http://www.spoj.com/problems/MMMGAME/
Problema 4.(MISERE NIM)
http://acm.mipt.ru/judge/problems.pl?problem=103&CGISESSID=6a967c679207f463295b609dfa198f10

Soluções:
Problema 1:

```
#include<stdio.h>
int main()
{
	int n;
	scanf("%d",&n);
	int soma=0;
	for (int i=1;i<=n;i++)
		{
			int a;
			scanf("%d",&a);
			soma^=a;
```

```
		}
	if (soma==0) printf("Second wins.\n");
	else printf("First wins.\n");
	return 0;
}
```

Problema 2:

```
#include<stdio.h>
int main()
{
	int t;
	scanf("%d",&t);
	for (int i=1;i<=t;i++)
		{
			int n;
			int soma=0;
			scanf("%d",&n);
			for (int j=1;j<=n;j++)
				{
				int a;
				scanf("%d",&a);
				soma+=a/(j);
				}
			 if(soma%2==1) printf("ALICE\n");
			 else printf("BOB\n");
		}
	return 0;
}
```

Problema 3:

```
#include<stdio.h>
int main()
{
	int t;
	scanf("%d",&t);
	for (int i=1;i<=t;i++)
		{
		int n;
		scanf("%d",&n);
		int soma=0;
		int maior=0;
		for (int j=1;j<=n;j++)
			{
			int a;
			scanf("%d",&a);
			if (a>maior) maior=a;
			soma=soma^a;
			}
		if (maior==1)
```

```
                                        {
                                                if (soma==0) printf("John\n");
                                                else printf("Brother\n");
                                        }
                else{
                        if (soma==0) printf("Brother\n");
                        else printf("John\n");
                        }
                }
        return 0;
}
```

Problema 4: É exatamente igual ao problema 3.