

# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memset0**  
Proudly powered by **Typecho**

## BZOJ2410 Nim游戏

2019-01-04 | 题解

@GNAQ 给的博弈劲爆题，让人无言以对 233.

**$k > 2$**

对于这种情况，我们可以发现任何时候一定可以给空白格子染色。所以我们只需要判断剩余空白格子的奇偶性即可

**$k = 1$**

显然被已经染色的格子分隔出的连续空白格子是不会相互影响的，所以我们可以把他看做一个组合 Nim 游戏，最后判断求一下每个游戏的 SG 函数异或和。

首先你要知道 mex 函数是什么，这里不再赘述，以下的打表程序中用到的实现是：

```
int mex(const std::set<int> &set) {  
    if (!set.size() || *set.begin()) return 0;  
    for (auto i = set.begin(), j = ++set.begin(); j != set.end(); i = j, j++)  
        if ((*i) + 1 != (*j)) return (*i) + 1;  
}
```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memset0**

Proudly powered by **Typecho**

```
return (*--set.end()) + 1;
}
```

此时题目可以转换为每次染色的格子必须与已经染色的格子不相邻。设  $sg1(x)$  为有连续  $x$  个空白格子可以放的 SG 值，则可以推出式子（注意边界情况）：

$$sg1(i) = mex(sg1(j) \oplus sg1(i - j - 3) \mid 1 \leq j \leq i - 4, sg1(i - 2), sg1(i - 3))$$

写个程序打表：

```
sg[0] = 0, sg[1] = 1, sg[2] = 1, sg[3] = 2;
for (int i = 4; i <= n; i++) {
    set.clear(), set.insert(sg[i - 2]), set.insert(sg[i - 3]);
    for (int j = 1; i - j - 3 >= 1; j++)
        set.insert(sg[j] ^ sg[i - j - 3]);
    sg[i] = mex(set);
}
```

跑出来以后，你可能会大叫这是什么 \*\* 玩意儿，下面是表：

```
1 1 2 0 3 1 1 0 3 3 2 2 4 0 5 2 2 3 3 0 1 1 3 0 2 1 1 0 4 5 2 7 4 0
1 1 2 0 3 1 1 0 3 3 2 2 4 4 5 5 2 3 3 0 1 1 3 0 2 1 1 0 4 5 3 7 4 8
1 1 2 0 3 1 1 0 3 3 2 2 4 4 5 5 9 3 3 0 1 1 3 0 2 1 1 0 4 5 3 7 4 8
1 1 2 0 3 1 1 0 3 3 2 2 4 4 5 5 9 3 3 0 1 1 3 0 2 1 1 0 4 5 3 7 4 8
1 1 2 0 3 1 1 0 3 3 2 2 4 4 5 5 9 3 3 0 1 1 3 0 2 1 1 0 4 5 3 7 4 8
```

可以发现每 34 位一循环，除了第一行和第二行有些特例（我才不会告诉你我是因为当初没仔细看后几位的差别才找到这个循环节的...）



# memseto's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memseto**  
Proudly powered by **Typecho**

$k = 2$

同理，我们可以这样假设 SG 函数（请注意这里的定义与  $k = 1$  有略微的差别）：

- $sg2(x, 0)$  表示连续  $x$  个空白格子且两端格子相同
- $sg2(x, 1)$  表示连续  $x$  个空白格子且两端格子不同
- $sg2(x, 2)$  表示连续  $x$  个空白格子且一端是已经染色的格子，另一端是边缘
- $sg2(x, 3)$  表示连续  $x$  个空白格子且两端都是边缘

虽然最后一种情况，但是数据里并没有，你不判断也没什么关系。

然后推一波式子（懒得写  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  了，直接看下面的打表程序）

```
sg[0][0] = sg[0][1] = sg[0][2] = sg[0][3] = sg[1][1] = 0;
sg[1][0] = sg[1][2] = sg[1][3] = 1;
for (int i = 2; i <= n; i++) {
    set.clear(), set.insert(sg[i - 1][1]);
    for (int j = 1; i - j - 1 >= 1; j++) {
        set.insert(sg[j][0] ^ sg[i - j - 1][0]);
        set.insert(sg[j][1] ^ sg[i - j - 1][1]);
    }
    sg[i][0] = mex(set);

    set.clear(), set.insert(sg[i - 1][0]);
    for (int j = 1; i - j - 1 >= 1; j++)
        set.insert(sg[j][0] ^ sg[i - j - 1][1]);
    sg[i][1] = mex(set);

    set.clear(), set.insert(sg[i - 1][2]), set.insert(sg[i - 1][0]), set.inse
    for (int j = 1; i - j - 1 >= 1; j++) {
        set.insert(sg[j][0] ^ sg[i - j - 1][2]);
```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memset0**  
Proudly powered by **Typecho**

```

        set.insert(sg[j][1] ^ sg[i - j - 1][2]);
    }
    sg[i][2] = mex(set);

    set.clear(), set.insert(sg[i - 1][2]);
    for (int j = 1; i - j - 1 >= 1; j++)
        set.insert(sg[j][2] ^ sg[i - j - 1][2]);
    sg[i][3] = mex(set);
}

```

可以发现：

- $sg2(x, 0) = 1$
- $sg2(x, 1) = 0$
- $sg2(x, 2) = x$
- $sg2(x, 3) = x \bmod 2$

综上所述即可写出程序。

代码：

```

// =====
//   author: memset0
//   date: 2019.01.04 10:26:33
//   website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
}
}

```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme [Ringo](#) by [memset0](#)  
Proudly powered by [Typecho](#)

```

    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10;
int T, n, m, a[N];

inline void finally(bool flag) {
    putchar(flag ? 'y' : 'n'), putchar('\n');
}

int sg1_1[] = {
    0, 1, 1, 2, 0, 3, 1, 1, 0, 3, 3, 2, 2, 4, 0, 5, 2, 2, 3, 3, 0, 1, 1, 3, 0
    0, 1, 1, 2, 0, 3, 1, 1, 0, 3, 3, 2, 2, 4, 4, 5, 5, 2, 3, 3, 0, 1, 1, 3, 0
}, sg1_2[] = {
    8, 1, 1, 2, 0, 3, 1, 1, 0, 3, 3, 2, 2, 4, 4, 5, 5, 9, 3, 3, 0, 1, 1, 3, 0
};

inline int sg1(int x) {
    if (x <= 0) return 0;
    if (x < 68) return sg1_1[x];
    return sg1_2[x % 34];
}

inline int sg2(int x, int y) {
    if (!x) return 0;
    if (y < 2) return y ^ 1;
    if (y == 3) return x & 1;
    return x;
}

void main() {

```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memset0**  
Proudly powered by **Typecho**

```
for (read(T); T--; ) {
    read(n), read(m);
    for (register int i = 1; i <= n; i++)
        read(a[i]);
    if (m == 1) {
        int cnt = 1, ans = 0;
        for (int i = 1; i <= n; i++)
            if (a[i]) {
                ans ^= sg1(cnt - 2);
                cnt = 0;
            } else ++cnt;
        ++cnt, ans ^= sg1(cnt - 2);
        finally(ans);
    } else if (m == 2) {
        int cnt = 0, lst = 0, ans = 0;
        for (register int i = 1; i <= n; i++)
            if (a[i]) {
                if (lst == 0) ans ^= sg2(cnt, 2);
                else if (lst == a[i]) ans ^= sg2(cnt, 0);
                else if (lst != a[i]) ans ^= sg2(cnt, 1);
                cnt = 0, lst = a[i];
            } else ++cnt;
        if (lst == 0) ans ^= sg2(cnt, 3);
        else ans ^= sg2(cnt, 2);
        finally(ans);
    } else {
        int cnt = 0;
        for (register int i = 1; i <= n; i++)
            cnt += !a[i];
        finally(cnt & 1);
    }
}

} signed main() { return ringo::main(), 0; }
```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme Ringo by memset0  
Proudly powered by Typecho

打表程序1：

```
// =====
//  author: memset0
//  date: 2019.01.04 09:37:28
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10;
int n, sg[N];
std::set <int> set;

int mex(const std::set <int> &set) {
    if (!set.size() || *set.begin()) return 0;
    for (auto i = set.begin(), j = ++set.begin(); j != set.end(); i = j, j++)
        if ((*i) + 1 != (*j)) return (*i) + 1;
    return (*--set.end()) + 1;
}

void main() {
    n = 200;
```



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme Ringo by memset0  
Proudly powered by Typecho

```
sg[0] = 0, sg[1] = 1, sg[2] = 1, sg[3] = 2;
for (int i = 4; i <= n; i++) {
    set.clear(), set.insert(sg[i - 2]), set.insert(sg[i - 3]);
    for (int j = 1; i - j - 3 >= 1; j++)
        set.insert(sg[j] ^ sg[i - j - 3]);
    sg[i] = mex(set);
}
for (int i = 1; i <= n; i++) {
    print(sg[i], ' ');
    if (i % 34 == 0) putchar('\n');
}

} signed main() { return ringo::main(), 0; }
```

打表程序2：

```
// =====
//  author: memset0
//  date: 2019.01.04 09:37:28
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
```





# memseto's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme [Ringo](#) by [memseto](#)  
Proudly powered by [Typecho](#)

```

}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10;
int n, sg[N][4];
std::set<int> set;

int mex(const std::set<int> &set) {
    if (!set.size() || *set.begin()) return 0;
    for (auto i = set.begin(), j = ++set.begin(); j != set.end(); i = j, j++)
        if ((*i) + 1 != (*j)) return (*i) + 1;
    return (*--set.end()) + 1;
}

void main() {
    n = 200;
    sg[0][0] = sg[0][1] = sg[0][2] = sg[0][3] = 0;
    sg[1][0] = 1, sg[1][1] = 0, sg[1][2] = 1, sg[1][3] = 1;
    for (int i = 2; i <= n; i++) {
        set.clear(), set.insert(sg[i - 1][1]);
        for (int j = 1; i - j - 1 >= 1; j++) {
            set.insert(sg[j][0] ^ sg[i - j - 1][0]);
            set.insert(sg[j][1] ^ sg[i - j - 1][1]);
        }
        sg[i][0] = mex(set);

        set.clear(), set.insert(sg[i - 1][0]);
        for (int j = 1; i - j - 1 >= 1; j++)
            set.insert(sg[j][0] ^ sg[i - j - 1][1]);
        sg[i][1] = mex(set);

        set.clear(), set.insert(sg[i - 1][2]), set.insert(sg[i - 1][0]), set.
        for (int j = 1; i - j - 1 >= 1; j++) {
            set.insert(sg[j][0] ^ sg[i - j - 1][2]);
            set.insert(sg[j][1] ^ sg[i - j - 1][2]);
        }
        sg[i][2] = mex(set);
    }
}

```



# memseto's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memseto**  
Proudly powered by **Typecho**

```
set.clear(), set.insert(sg[i - 1][2]);
for (int j = 1; i - j - 1 >= 1; j++)
    set.insert(sg[j][2] ^ sg[i - j - 1][2]);
sg[i][3] = mex(set);
}
for (int i = 1; i <= n; i++)
    printf("%d %d %d %d\n", sg[i][0], sg[i][1], sg[i][2], sg[i][3]);
}

} signed main() { return ringo::main(), 0; }
```

博弈

SG 函数

## 仅有一条评论



**AThinker**

March 12th, 2019 at 07:49 pm

资瓷！

回复

用户名

邮箱

网址（选填）

可以在这里写评论哦 ~



# memseto's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memseto**  
Proudly powered by **Typecho**

提交评论

**BZOJ4179 B**  
上一篇 «

**LOJ6514 「雅礼集训 2018 Day10」**  
**文明**  
» 下一篇

© 2017 - 2019 **memset0** 的博客.  
**浙ICP备19006255号-1**  
97687 visits · 24756 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

