

1325: 循环比赛日程表

题目描述

设有 N 个选手进行循环比赛，其中 $N=2^M$ ，要求每名选手要与其他 $N-1$ 名选手都赛一次，每名选手每天比赛一次，循环赛共进行 $N-1$ 天，要求每天没有选手轮空。

输入

输入： M 。

输出

表格形式的比赛安排表。一行各数据间用一个空格隔开。

输入样例

3

输出样例

```
1 2 3 4 5 6 7 8
2 1 4 3 6 5 8 7
3 4 1 2 7 8 5 6
4 3 2 1 8 7 6 5
5 6 7 8 1 2 3 4
6 5 8 7 2 1 4 3
7 8 5 6 3 4 1 2
8 7 6 5 4 3 2 1
```

分析

这道题需要仔细观察最终的输出图像

	0	1	2	3	4	5	6	7
0		1	2					
1		2	1					
2								
3								
4								
5								
6								
7								

	0	1	2	3	4	5	6	7
0		1	2	3	4			
1		2	1	4	3			
2		3	4	1	2			
3		4	3	2	1			
4								
5								
6								
7								

	0	1	2	3	4	5	6	7
0		1	2	3	4	5	6	7
1		2	1	4	3	6	5	8
2		3	4	1	2	7	8	5
3		4	3	2	1	8	7	6
4		5	6	7	8	1	2	3
5		6	5	8	7	2	1	4
6		7	8	5	6	3	4	1
7		8	7	6	5	4	3	2

怎么样，看到这里是不是豁然开朗了？

总结规律如下：

1、每幅图中左上与左下都是完全对应的，右上与右下也是完全对应的，因此我们只要计算出左上和右上就能够直接映射出左下和右下。

2、每幅图中的右上（红色部分）又等于左上的值（绿色部分）加上规模，规模等于1,2,4,8...
...

因此，当我们拿到一个指定规模的数据时，我们可以将它除以2进行缩小，例如8→4，4→2，2→1，直到0，当规模为0的时候，我们的基础数据很明显就是1。

计算完基础规模后，我们再去按照规律2计算右上，之后再映射左下和右下即可。

编码

```
#include<bits/stdc++.h>

using namespace std;
const int maxN = 1000;
//定义矩阵值
int Rec[maxN][maxN];

void Count(int size) {
    if (size == 0) {
        Rec[size][size] = 1;
        return;
    }
    //计算左上角;
    Count(size / 2);
    //计算右上角
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            //用0加上规模，即是右边数据的位置
            Rec[i][j + size] = Rec[i][j] + size;
        }
    }
    //映射左下和右下
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            //左下
            Rec[i + size][j] = Rec[i][j + size];
            //右下
            Rec[i + size][j + size] = Rec[i][j];
        }
    }
}
```

```
}

void Printf(int limit) {
    for (int i = 0; i < limit; ++i) {
        for (int j = 0; j < limit; ++j) {
            printf("%d ", Rec[i][j]);
        }
        printf(("\\n"));
    }
}

int main() {
    int m;
    cin >> m;
    int limit = pow(2, m);
    Count(limit);
    Printf(limit);
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

