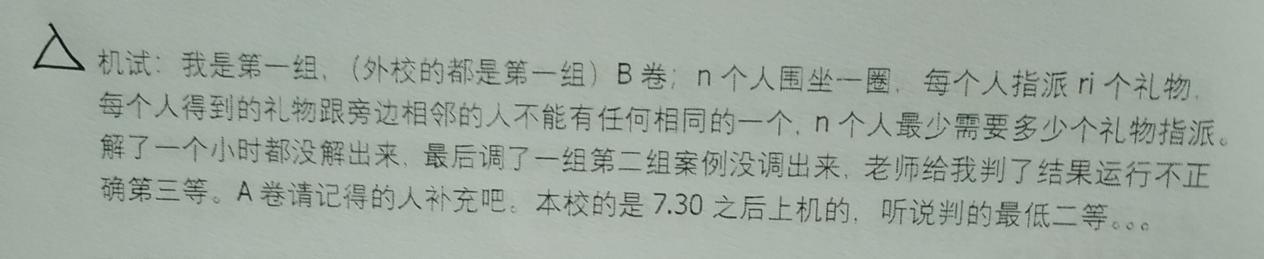
# 分发礼物问题

## 分发礼物



我的思路：

将n个人的礼物用数字表示，数字不同表示不同的礼物，n个人的礼物用a[n]数组来存储，从a[0]~a[n-1]，其中，a[0] = 1,a[1] = 2,a[n-1]=3，即表示开头第一个人的礼物与左右两边都不一样

其他的数组元素满足：a[i] != a[(i-1+n)%n] && a[i] != a[(i+1+n)%n] 从1~n不断尝试，知道得到满足条件的数组组合，最后遍历数组，找出其中最大的数字即为n个人最少需要的礼物数量

递归设计：好像是递归错误了，找不到方法回退

```c++

#include <cstdio>

#include <iostream>

#include <algorithm>

using namespace std;

const int MAXN = 200;

int a[MAXN];

bool present(int n,int k,int w){//a[k]位,即第k+1个人尝试从1~n赋值,w为a[k]的值

if (w > n || w < 1 || k < 2) {

return false;

}

if (k > n -1 ) {

for (int i = 0; i < n; ++i) {//如果不是合法情况，则返回false

if (!(a[i] != a[(i - 1 + n) % n] && a[i] != a[(i + 1 + n) % n])) {

return false;

}

}

for (int i = 0; i < n; ++i) {//合法情况，输出满足的结果

printf("%d",a[i]);

}

printf("\n");

return true;

}

a[k] = w;

present(n, k ,w + 1);

present(n, k + 1 ,w);

present(n, k-1, w);

present(n, k, w - 1);

return false;

}

int main(){

int n;

printf("有几个小朋友需要礼物？");

scanf("%d",&n);

a[0] = 1;

a[1] = 2;

a[n-1] = 3;

if (present(n, 2, 1)) {

printf("YES");

}

return 0;

}

```