

A Naïve 全排列

分析

用递归实现。以下是一种思路

首先声明一个 n 长度的空数组，用递归填入数字

递归：

第一层：

传入参数：空数组，当前长度=0，总长度= n

循环， i 从1到 n ，如果 i 没有重复的话就填入 i ，并调用下一层，传入参数：有一项的数组，当前长度= $0+1=1$ ，总长度= n

第二层：

传入参数：有一项的数组，当前长度=1，总长度= n

循环， i 从1到 n ，如果 i 没有重复的话就填入 i ，并调用下一层，传入参数：有两项的数组，当前长度= $1+1=2$ ，总长度= n

...

第 n 层： 传入参数：有 $n-1$ 项的数组，当前长度= $n-1$ ，总长度= n

循环， i 从1到 n ，如果 i 没有重复的话就填入 i ，并调用下一层，传入参数：有 n 项的数组，当前长度= n ，总长度= n

第 $n+1$ 层：

判断发现当前长度=总长度= n ，因此中止递归，输出数组。 返回。

```
#include <stdio.h>

int hasSame(int *base, int l, int num){//判断新的数是否重复
    int i;
    for(i = 0; i < l; i++){
        if(base[i] == num) return 1;
    }
    return 0;
}

void prtarr(int *base, int l){//输出数组
    int i;
    for(i = 0; i < l; i++){
        printf("%d ", base[i]);
    }
    putchar('\n');
}

void arr(int *base, int l, int n){
    if(l == n){//终止条件: 当前长度=总长度
        prtarr(base, l);
        return;
    }
    int i;
    for(i = 1; i <= n; i++){
        if(!hasSame(base, l, i)){//判断i有没有重复
            base[l] = i;
            arr(base, l+1, n);//递归
        }
    }
}

int main(){
    int n;
    int base[30];
    scanf("%d", &n);
    arr(base, 0, n);
}
```