## 1.排列和组合的实现

/\*

功能：

在size范围内找出N个数的所有排列

参数解释：

arr为一个数组指针

N为要选择的个数

index为当前arr里面数的个数

size是所选择的范围

例子：

int arr[3];

pailie(arr,3,0,9);

在0-8中选出3个数的所有排列方式

\*/

int pailie(int \*arr,int N,int index,int size)

{

int i,j,flag=1,k;

if(index==N)

{

for(k=0;k<N;k++)

{

printf("%d ",\*(arr+k));

}

printf("\n");

return 0;

}

for(i=0;i<size;i++)

{

flag=1;

for(j=0;j<index;j++)

{

if(i==\*(arr+j))//在这里或上一个i<\*(arr+j)就能实现组合

{

flag=0;

break;

}

}

if(flag==1)//说明i是没被选过的

{

\*(arr+index)=i;

//**重要：**递归调用参数传递都不是修改的数据本身，只是修改的有相同的值另一个地址上，要永久改变值就用指针操作，传递指针。

//**错误想法：**index++;这里不能使用index++，因为index++改变了index的值，到时候返回上一级时index不会减小，而index+1没改变index的值，在递归调用时都是传递的值的副本，每层的index的地址都不一样，之时这里应该在返回时把index值变回去

pailie(arr,N,index+1,size);

//可以在这里index--；

}

}

}

## DSF算法

回溯法（探索与回溯法）是一种选优搜索法，又称为试探法，按选优条件向前搜索，以达到目标。但当探索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择，这种走不通就退回再走的技术为回溯法，而满足回溯条件的某个状态的点称为“回溯点”。

int check(参数)

{

if(满足条件)

return 1;

return 0;

}

void dfs(int step)

{

判断边界

{

相应操作

}

尝试每一种可能

{

满足check条件

标记

继续下一步dfs(step+1)

恢复初始状态（回溯的时候要用到）

}

}

## 模型过拟合

含义：过拟合就是训练出来的模型在训练集上表现很好，但是在测试集上表现较差的一种现象

原因：1.数据量太小，未包含真实模型的全部信息

2.训练数据中有错误的数据，误导模型错误发展。

## 4.为什么要在神经网络里引入非线性函数

如果不用激活函数（其实相当于激活函数是f(x) = x），在这种情况下你每一层输出都是上层输入的线性函数，很容易验证，无论你神经网络有多少层，输出**都是输入的线性组合**，与没有隐藏层效果相当。引入非线性函数作为激活函数，这样深层神经网络就有意义了（不再是输入的线性组合，可以逼近任意函数）

## **5.rand35()实现rand47()**

思想：扩大概率范围，然后截取目标范围

int rand47()

{

int x;

while(x=rand35()\*35+rand35() > 47)

x=rand47();

return x;

}