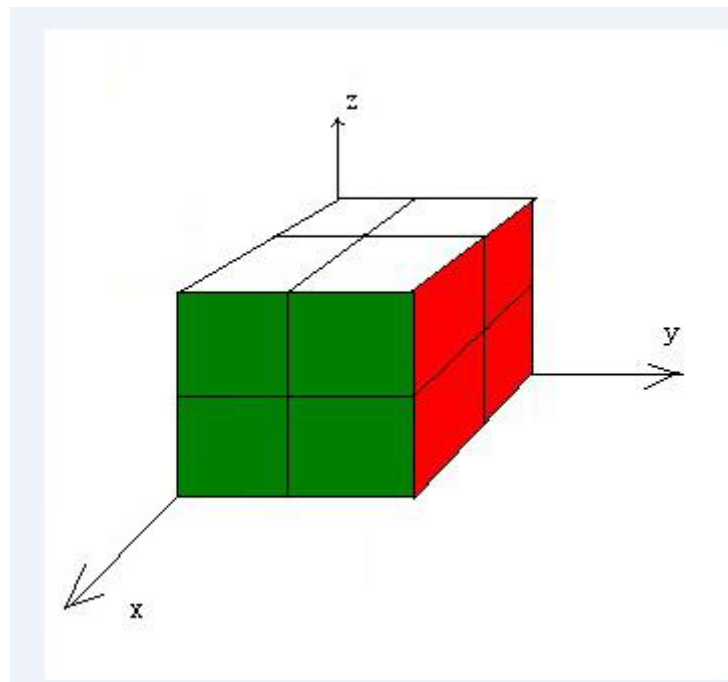


二阶魔方（Pocket Cube）算法

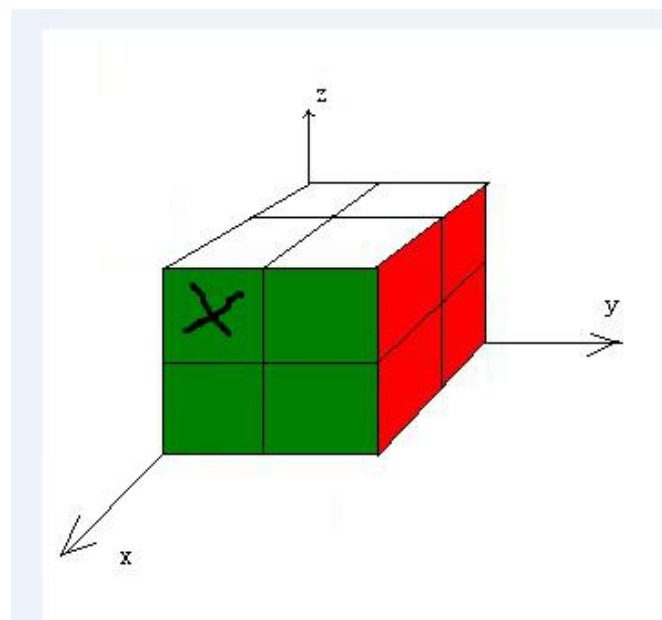
1 代码

1.1 数据结构

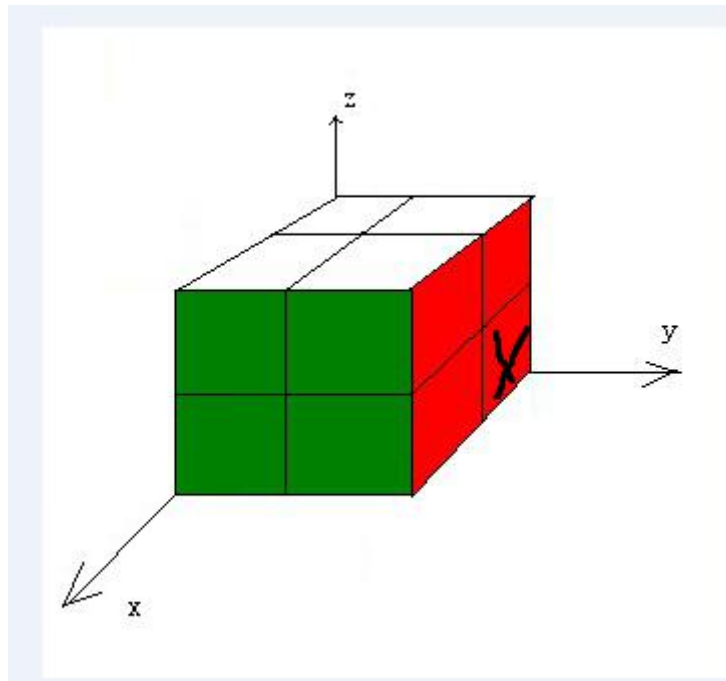


我们先定义颜色，0-5 分别表示橙蓝红绿黄白六种颜色。四维数组 `cube[2][2][2][6]` 用来存储 8 个方块共 24 面的颜色。其中前三个维度为方块的坐标，分别为 (x, y, z) ，第四个维度为该块三个面的位置，我们对第四个维度规定如下：0 表示朝前的面，1 表示朝右的面，2 表示朝后的面，3 包表示朝左的面，4 表示朝上的面，5 表示朝下的面。

例如，下图中打叉的面可表示为 `cube[1][0][1][0]=4`。



下图中打叉的面可表示为 `cube[1][1][0][1]=2`。



1.2 函数

```
int main()//主函数
int InitCube()//初始化函数，新建四维数组并从键盘输入魔方的初始状态。
int FirstStep()//还原魔方的第一步
int SecondStep()//还原魔方的第二步
int ThirdStep()//还原魔方的第三步
int Rotate(int direction)//对魔方进行旋转操作，direction 为 0、1、2、3 分别
表示左、上、右、下
int PrintCube()//将魔方当前的状态打印出来
int JudgeSolve()//检验魔方是否已经复原
```