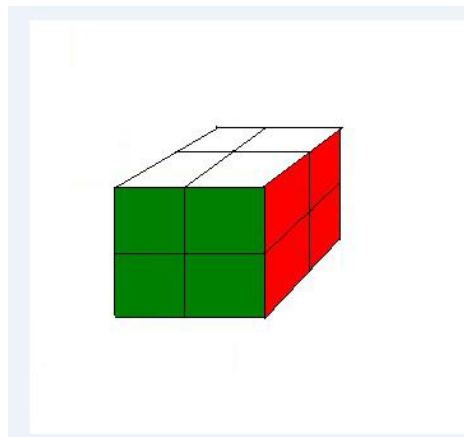


二阶魔方（Pocket Cube）算法

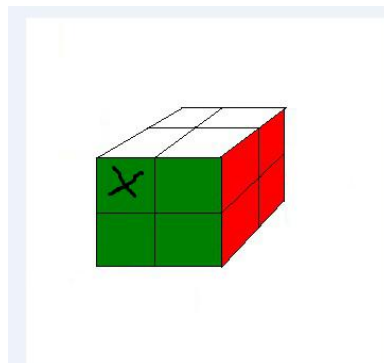
1 代码

1.1 数据结构

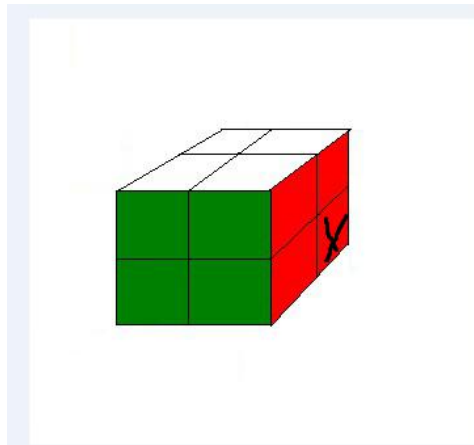


我们先定义颜色, 0-5 分别表示橙蓝红绿黄白六种颜色。二维数组 `cube[6][4]` 用来存储 8 个方块共 24 面的颜色。其中第一个维度用来表示 6 个面, 0-5 分别为前、右、后、左、上、下, 第二个维度表示该面四个小块, 按照从左到右从上到下依次为 0、1、2、3。

例如, 下图中打叉的面可表示为 `cube[0][0]=3`。



下图中打叉的面可表示为 `cube[1][3]=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
```