

地图四染色问题

问题描述:

给地图上的七个地方用四中颜色上色.用1表示两地相邻,用0表示两地不相邻.相邻两地颜色不能相同.

二维数组a[i][j]

	A	B	C	D	E	F	G
A	0	1	1	1	1	1	0
B		0	0	0	0	1	0
C			0	1	1	0	0
D				0	1	1	0
E					0	1	0
F						0	0
G							0

用1, 2, 3, 4 分别代表不同的颜色.

思路

想到可以用栈来实现穷举.

一维数组b[j]

A	B	C	D	E	F	G	H

先将两个颜色入栈, 栈顶指针指向B. 即 $p = 1$;

A	B	C	D	E	F	G	H
1	1						

再判断这样的配色是否合法

```
while (i = 0; i <= p; i++){
    while (j = i+1; j <= p; j++){
        if (a[i][j] == 1 && b[j] == b[i]) return 0;
        //Illegal
    }
}
return 1; //Legal
```

很不幸, 这样的配色在第一轮就被查出不合法了. 所以要将B的颜色改为2.

如果一轮检查发现(比如)E取1 2 3 4 都不满足条件怎么办? 那就说明D的配色出现了问题. 此时栈顶指针指向E, 那么需要将E出栈, 将D的颜色加一, 再将E取1, 2, 3, 4, 判断合理性. 如果D的其他取值也不能满足呢? 就说明C的取值不合理. 需要改变C的值. 以此类推.

如此反复, 地图四染色问题就解决啦.