**分析：Properties of BFS（广度优先搜索）**

**Completeness（完备性）**：Yes

在树上或者图中只要有从起点到终点的路径，那么通过bfs我们一定能找到这条路径，所以bfs是具有完备性的(Completeness)。但是，如果每一个点的子节点不是有限个数的话就是No，因为有无限子节点的话是展开不完的

**Optimality（最优性）：**

这个不能笼统而论，对于在树上的bfs，optimality一定是yes，因为在路上，两点之间只有一条路径。

对于图，只有当满足随着深度的加深，路径的长度是非递减的时候才满足最优性，

通常来说是不满足的。

**Time complexity（时间复杂度）**：O（b的d次方）；b（每一层的节点数），d（树的深度）

1+b+b的2次方+……+b的d次方=O(b的d次方)

**Space complexity（空间复杂度）**：O（b的d次方），因为在过程中要存储每一个结点

**四皇后问题(直接暴力剪树枝dfs)：**

~~其实对于四皇后问题的状态数太少，剪枝都没有必要~~

**Code(c++):**

//四皇后问题

#include<cstdio>

#include<iostream>

using namespace std;

int ans;//ans是可行的方案数

int lie[10];

int le[10];

int ri[10];

int aaa[5][5];

void dfs(int t)

{

if(t==5){

ans++;

printf("第%d种:\n",ans);

for(int i=1;i<=4;i++){

for(int j=1;j<=4;j++)

printf("%d ",aaa[i][j]);

printf("\n\n");

}

return;

}

for(int i=1;i<=4;i++){

if(lie[i]==0&&le[t+i]==0&&ri[t-i+5]==0){

lie[i]=1;

le[t+i]=1;

ri[t-i+5]=1;

aaa[t][i]=1;

dfs(t+1);

aaa[t][i]=0;

lie[i]=0;

le[t+i]=0;

ri[t-i+5]=0;

}

}

}

int main()

{

dfs(1);

printf("方案数 ：%d\n",ans);

return 0;

}

**运行结果：**

