人工智能实验一

1. 实验题目

用深度优先遍历求解汉诺塔的问题。

1. 实验分析
2. 汉诺塔问题：

n个盘子和3根柱子：A(源)、B(备用)、C(目的)，盘子的大小不同且中间有一孔，可以将盘子“串”在柱子上，每个盘子只能放在比它大的盘子上面。起初，所有盘子在A柱上，问题是将盘子一个一个地从A柱子移动到C柱子。移动过程中，可以使用B柱，但盘子也只能放在比它大的盘子上面。

1. 限制条件：

1.在小圆盘上不能放大圆盘。

2.在三根柱子之间一回只能移动一个圆盘。

3.只能移动在最顶端的圆盘。

3. 深度优先搜索属于图算法的一种，是一个针对图和树的遍历算法，英文缩写为DFS即Depth First Search。深度优先搜索是图论中的经典算法，利用深度优先搜索算法可以产生目标图的相应拓扑排序表，利用拓扑排序表可以方便的解决很多相关的图论问题，如最大路径问题等等。一般用堆数据结构来辅助实现DFS算法。其过程简要来说是对每一个可能的分支路径深入到不能再深入为止，而且每个节点只能访问一次。

三.实验步骤

1、对于每一个状态(pegA, pegB, pegC)，由于要寻找其解，放入当前步骤（搜索层）问题表（队列堆栈皆可）。最开始初始化时表中只有一个元素，即最初状态。

2、维护一个搜索集set。对于状态(pegA, pegB, pegC)，如果已在搜索中，则该分支陷入重复，放弃该分支搜索；否则加入搜索集。

3、遍历当前问题表求解。如果有某一状态满足终止条件，那么当前步骤数即为所求，返回；如果没有，则展开得到更深层次的问题表，步骤数++。返回步骤1

四．实验内容

代码：#include<iostream>

#include<algorithm>

#include<cstdio>

#include<cstdlib>

#include<cmath>

#include<cstring>

#include<vector>

#include<stack>

#include<map>

#define maxn 70

#define lson step<<1

#define rson step<<1|1

using namespace std;

bool dfs(int n,int A[],int B[],int C[])

{

if(n==0)

return true;

if(B[0]==n)

return false;

else if(A[0]==n)

{

return dfs(n-1,A+1,C,B);

else if(C[0]==n)

{

return dfs(n-1,B,A,C+1);

}

return true;

}

int main()

{

int T,n,m,p,q;

cin>>T;

while(T--)

{

int A[maxn],B[maxn],C[maxn];

cin>>n;

cin>>m;

for(int i=0;i<m;i++)

{

cin>>A[i];

}

cin>>p;

for(int i=0;i<p;i++)

{

cin>>B[i];

}

cin>>q;

for(int i=0;i<q;i++)

{

cin>>C[i];

}

if(dfs(n,A,B,C)==true)

{

cout << "true\n";

}

else

cout << "false\n";

}

return 0;

}