人工智能实验六

171491218刘晓萌

1. 实验题目

迭代加深算法

1. 实验分析

迭代加深搜索（Iterative Deepening Depth-First Search, IDDFS）经常用于理论上解答树深度上没有上界的问题，这类问题通常要求出满足某些条件时的解即可

1. 实验内容

1.以埃及函数举例：要求将一个分数a/b分解成为若干个形如1/d的加数之和，而且加数越少越好，如果加数个数相同，那么最小的分数越大越好。

2.实验代码：

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<algorithm>

#include<string>

#include<sstream>

#include<set>

#include<vector>

#include<stack>

#include<map>

#include<queue>

#include<cstdlib>

#include<cstdio>

#include<cstring>

#include<cmath>

using namespace std;

typedef long long LL;

int maxd;

int a, b;

const int maxn = 1000;

int ans[maxn], v[maxn];

int gcd(int a, int b){

return b == 0 ? a : gcd(b, a%b);

}

int get\_first(int a, int b) //找到1/c≤a/b时最小的c

{

int c = 1;

while (b > a\*c)c++;

return c;

}

bool better(int d) //比较深度为d时，现在找到的解是不是更优的

{

for (int i = d; i >= 0; i--)

if (v[i] != ans[i]) {

return ans[i] == -1 || v[i] < ans[i]; //两种情况下说明当前更优：(1)此时尚未找到过解； (2)当前的分母小于原来的分母，说明当前的分数比原来的更大，符合题意要求

}

return false;

}

bool dfs(int d, int from, LL aa, LL bb) //当前深度为d,分母不能小于from,分数之和恰好是aa/bb

{

if (d == maxd) //到达了最后一层

{

if (bb%aa)

return false; //不能整除，说明最后一项不符合埃及分数的定义，失败退出

v[d] = bb / aa;

if (better(d))memcpy(ans, v, sizeof(LL)\*(d + 1)); //当前找到的解是更优的，更新ans

return true;

}

bool ok = false;

from = max(from, get\_first(aa, bb)); //更新from

for (int i = from;; i++) //枚举分母

{

if (bb\*(maxd + 1 - d) <= i\*aa) break; //利用乐观估价函数来剪枝，从当前深度d到达maxd一共有maxd-d+1项，如果(maxd-d+1)\*(1/i)还凑不够aa/bb，需要剪枝

v[d] = i;

LL b2 = bb\*i; //计算aa/bb-1/i，通分后，分母是bb\*i，分母是aa\*i-bb

LL a2 = aa\*i - bb;

LL g = gcd(a2, b2); //计算分子，分母的最大公约数，便于约分

if (dfs(d + 1, i + 1, a2 / g, b2 / g))ok = true;

}

return ok;

}

int main(){

int ok = 1;

while (scanf("%d%d", &a, &b)) //输入分数a/b

{

for (maxd = 1;; maxd++)

{

memset(ans, -1, sizeof(ans));

if (dfs(0, get\_first(a, b), a, b)){ ok = 1; break; }

}

printf("%d/%d=", a, b);

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

if (ans[i]>0)

printf("%s1/%d", i == 0 ? "" : "+", ans[i]);

else { printf("\n"); break; }

}

return 0;

}

3.实验截图：

