**实验六 迭代加深搜索**

**一、实验目的**

理解迭代加深搜索算法的基本操作和思想。

**二、实验分析**

1.迭代加深搜索，实质上就是限定下界的深度优先搜索。即首先允许深度优先搜索K层搜索树，若没有发现可行解，再将K+1后重复以上步骤搜索，直到搜索到可行解。

2.在迭代加深搜索的算法中，连续的深度优先搜索被引入，每一个深度约束逐次加1，直到搜索到目标为止。

3.迭代加深搜索算法就是仿广度优先搜索的深度优先搜索。既能满足深度优先搜索的线性存储要求，又能保证发现一个最小深度的目标结点。

4.从实际应用来看，迭代加深搜索的效果比较好，并不比广度优先搜索慢很多，但是空间复杂度却与深度优先搜索相同，比广度优先搜索小很多。

**三、实验内容**

已知一个数列a0，a1，……，am（其中a0=1，am=n，a0<a1<……<am-1<am）。

对于每个k，需要满足ak=ai+aj（0 ≤ i，j ≤ k-1，这里i与j可以相等）。

现给定n的值，要求这个数列每一项的值。

**源代码：**

#include<cstdio>

#include<iostream>

#include<cstring>

#include<algorithm>

using namespace std;

int n,ans[105];

bool f;

void dfs(int x,int de)

{

if(f) //如果为真就结束搜索

return;

if(x==de) //判断是否搜索完毕

{

if(ans[x]==n)

f=1;

return;

}

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

{

for(int j=i;j<=x;j++)

if(ans[i]+ans[j]>ans[x]&&ans[i]+ans[j]<=n)

{

int sum=ans[i]+ans[j];

for(int k=x+2;k<=de;k++)

sum\*=2;

if(sum<n)

continue;

ans[x+1]=ans[i]+ans[j];

dfs(x+1,de);

if(f)

return;

}

}

}

int main()

{

while(scanf("%d",&n)!=-1) //输入有误就无法进行

{

if(n==0) //如果输入为0，则程序结束

return 0;

memset(ans,0,sizeof(ans));//初始化ans[]

f=0;

ans[0]=1;

int m=1,depth=0;

while(m<n) //求所需深度

{

m\*=2;

depth++;

}

while(1) //进行搜索

{

dfs(0,depth);

if(f)

break;

depth++;

}

printf("%d",ans[0]);

for(int i=1;i<=depth;i++) //输出数列

printf(" %d",ans[i]);

printf("\n");

}

}

**运行结果：**

