

分支限界法求 0-1 背包问题实验程序以及代码（ C++ ）

本程序中（规定物品数量为 3，背包容量为 30，输入为 6 个数，前 3 个为物品重量，后 3 个数为物品价值）：

代码：

```
#include<iostream>

#include<stack>

using namespace std;

#define N 100

class
HeapNode // 定义 HeapNode 结点类
{
public:
double
upper,price,weight; //upper 为结点的价值上界， price 是结点所对应的价值， weight 为结点所相应的重量
int level,x[N]; // 活节点在子集树中所处的层序号
};

double MaxBound(int i);

double Knap();

void AddLiveNode(double up,double cp,double cw,bool ch,int level);

stack<HeapNode>

High; // 最大队 High

double w[N],p[N]; // 把物品重量和价值定义为双精度浮点数

double cw,cp,c=30; //cw 为当前重量， cp 为当前价值，定义背包容量为 30

int n=3; // 货物数量为 3

int main()

{

cout<<" 请按顺序输入 3 个物品的重量：（按回车键区分每个物品的重量）" <<endl;

int i;

for(i=1;i<=n;i++)

cin>>w[i]; // 输入 3 个物品的重量
```

```
cout<<" 请按顺序输入  3 个物品的价值：（按回车键区分每个物品的价值）" <<endl;
```

```
for(i=1;i<=n;i++)
```

```
cin>>p[i];  // 输入 3 个物品的价值
```

```
cout<<" 最大价值为：  ";
```

```
cout<<Knap()<<endl; // 调用 knap 函数 输出最大价值
```

```
return 0;
```

```
}
```

```
double MaxBound(int j) //MaxBound  函数求最大上界
```

```
{
```

```
double
```

```
left=c-cw,b=cp;  // 剩余容量和价值上界
```

```
while(j<=n&&w[j]<=left)  // 以物品单位重量价值递减装填剩余容量
```

```
{
```

```
left-=w[j];
```

```
b+=p[j];
```

```
j++;
```

```
}
```

```
if(j<=n)
```

```
b+=p[j]/w[j]*left;  // 装填剩余容量装满背包
```

```
return b;
```

```
}
```

```
void AddLiveNode(double up,double cp,double cw,bool ch,int lev)
```

```
//将一个新的活结点插入到子集数和最大堆      High 中
```

```
{
```

```
HeapNode be;
```

```
be.upper=up;
```

```
be.price=cp;
```

```
be.weight=cw;
```

```
be.level=lev;
```

```
if(lev<=n)
```

```
High.push(be); // 调用 stack 头文件的 push 函数
```

```
}
```

```

double Knap() // 优先队列分支限界法，返回最大价值，    bestx 返回最优解

{ int i=1; cw=cp=0; double

bestp=0,up=MaxBound(1); //    调用 MaxBound  求出价值上界，    best  为最优值

while(1)    // 非叶子结点

{ double wt=cw+w[i];

if(wt<=c) // 左儿子结点为可行结点

{ if(cp+p[i]>bestp) bestp=cp+p[i];

AddLiveNode(up,cp+p[i],cw+w[i],true,i+1);

}

up=MaxBound(i+1);

if(up>=bestp) // 右子数可能含最优解

AddLiveNode(up,cp,cw,false,i+1);

if(High.empty()) return bestp;

HeapNode node=High.top(); //    取下一扩展结点

High.pop(); cw=node.weight; cp=node.price; up=node.upper;

i=node.level;

}

}

```

输出结果为：

```

请按顺序输入3个物品的重量：（按回车键区分每个物品的重量）
16
15
15
请按顺序输入3个物品的价值：（按回车键区分每个物品的价值）
45
25
25
最大价值为：50
Press any key to continue

```