

0-1 背包 - 动态规划

作者: [Saber](#)

原文链接: <https://ld246.com/article/1517131738346>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

0-1背包-动态规划

```
include "stdafx.h"
#include "iostream"
#define CAP 1000 //背包重量上限
#define num 100 //物品数量上限
using namespace std;

int w[num]; //重量
int v[num]; //价值
int p[num][num]; //动态规划数组
int x[num]; //最优解数组
void Knaspack(int c,int n)
{
    int wmax=min(w[n-1],c);
    for(int i=0;i<=wmax;i++)
    {
        p[n][i]=0;
    }
    for(int i=w[n];i<=c;i++)
    {
        p[n][i]=v[n];
    }
    for(int i=n-1;i>1;i--)
    {
        wmax=min(w[i],c);
        for(int j=0;j<=wmax;j++)
        {
            p[i][j]=p[i+1][j];
        }
        for(int j=w[i];j<=c;j++)
        {
            p[i][j]=max(p[i+1][j],p[i+1][j]-w[i]+v[i]);
        }
    }
    p[1][c]=p[2][c];
    if(c>=w[1])
        p[1][c]=max(p[2][c],p[2][c-w[1]]+v[1]);
}

void traceback(int c,int n)
{
    for(int i=1;i<n;i++)
    {
        if(p[i][c]==p[i+1][c])
            x[i]=0;
        else
        {
            x[i]=1;
            c=c-w[i];
        }
    }
    x[n]=(p[n][c])?1:0;
}
```

```
}  
}
```

```
int _tmain(int argc, _TCHAR* argv[])  
{  
    w[1]=2;w[2]=1;w[3]=3;w[4]=2;  
    v[1]=12;v[2]=10;v[3]=20;v[4]=15;  
    Knaspack(5,4);  
    traceback(5,4);  
    cout<<"最优值为: "<<p[1][5]<<endl;  
    cout<<"最优解为: {";  
    for(int i=1;i<4;i++)  
        cout<<x[i]<<" ";  
    cout<<x[4]<<"}"<<endl;  
    system("pause");  
    return 0;  
}
```