五大算法之贪心法

作者：yanghang

原文链接：https://ld246.com/article/1482223222258

来源网站：链滴

<p><span style="font-size: medium;">问题：一个旅行者有一个最多能用c公斤的背包，现在有n物品，每件的重量分别是w1，w2，...,wn,每件的价值分别为v1,v2,...,vn，若每种物品都可无限细分，旅行者能获得的最大总价值。</span></p>
<p><span style="font-size: medium;">抽象：组合出价值最大的指定重量的物品。</span></p>
<p><span style="font-size: medium;">思路：先求出每个物品的价值密度，先放入价值密度最大物品，再放入剩余物品价值密度最大的物品，依次进行，直到背包已满。</span></p>
<p><span>代码：</span></p>
<pre class="brush: java">package test;

import java.util.Arrays;

/**

- 贪心法求背包问题（可分割）
- @author yanghang
-

*/

public class Tanxin {

```
public static void main(String[] args) {
    /**
     * 初始化
     */
    // 背包的容量
    double c = 12;
    // 物品的数量
    int n = 5;
    // 物品重量数组
    double[] w = new double[n];
    // 物品价钱数组
    double[] v = new double[n];
    // 物品的重量  1,2,3,4,5
    for (int i = 0; i &lt; n; i++) {
        w[i] = i + 1;
    }
    // 物品的价值 5,4,3,2,1
    for (int i = 0; i &lt; n; i++) {
        v[i] = n - i;
    }

    /**
     * 求出单位重量价值r[i] = v[i] / w[i]
     *
     */
    double[] r = new double[n];
    int[] index = new int[n];
    for (int i = 0; i &lt; n; i++) {
        r[i] = (double) v[i] / (double) w[i];
        index[i] = i;
    }
    /**
     * 按照价值密度排序
     */
    double temp = 0;
    for (int i = 0; i &lt; n - 1; i++) {
        for (int j = i + 1; j &lt; n; j++) {
            if (r[i] &lt; r[j]) {
                temp = r[i];
                r[i] = r[j];
                r[j] = temp;
                temp = w[i];
                w[i] = w[j];
                w[j] = temp;
                temp = v[i];
                v[i] = v[j];
                v[j] = temp;
            }
        }
    }

    /**
     * 初始化解向量x[n],求解并打印解向量
     */
    double[] x = new double[n];
    for (int i = 0; i &lt; n; i++) {
        if (w[i] &lt;= c) {
            x[i] = 1;
            c = c - w[i];
        } else if (w[i] &gt; c) {
            x[i] = (double)c / w[i];
            c = 0;
            break;
        }
    }
    System.out.println("解向量是：" + Arrays.toString(x));


    /**
     * 根据解向量求出背包中存放物品的最大价值并打印
     */
    double maxValue = 0;
    for (int i = 0; i &lt; n; i++) {
        maxValue += v[i] * x[i];
    }
    System.out.println("背包中物品的最大价值为：" + maxValue);

}

}</pre>

<p><span>&nbsp;</span></p>
```

原文链接：五大算法之贪心法