测试报告

测试概要

标识符：bestv 测试内容：测试多维背包问题最大价值的正确性

测试结果及发现

测试用例1：

6 10 3800

100 600 1200 2400 500 2000

8 12 13 64 22 41

8 12 13 75 22 41

3 6 4 18 6 4

5 10 8 32 6 12

5 13 8 42 6 20

5 13 8 48 6 20

0 0 0 0 8 0

3 0 4 0 8 0

3 2 4 0 8 4

3 2 4 8 8 4

80 96 20 36 44 48 10 18 22 24

将以上数据存入txt文件中由Java程序文件流调用此txt文件实现测试数据输入

|  |  |
| --- | --- |
| 预期结果 | 实际结果 |
| 输出多维背包问题最大价值3800 | 和预期一致 |

测试用例2：

10 10 8706.1

600.1 310.5 1800 3850 18.6 198.7 882 4200 402.5 327

20 5 100 200 2 4 60 150 80 40

20 7 130 280 2 8 110 210 100 40

60 3 50 100 4 2 20 40 6 12

60 8 70 200 4 6 40 70 16 20

60 13 70 250 4 10 60 90 20 24

60 13 70 280 4 10 70 105 22 28

5 2 20 100 2 5 10 60 0 0

45 14 80 180 6 10 40 100 20 0

55 14 80 200 6 10 50 140 30 40

65 14 80 220 6 10 50 180 30 50

450 540 200 360 440 480 200 360 440 480

将以上数据存入txt文件中由Java程序文件流调用此txt文件实现测试数据输入

|  |  |
| --- | --- |
| 预期结果 | 实际结果 |
| 输出多维背包问题最大价值8706.1 | 和预期有一点误差 |

误差原因分析：由于代码实现数据整型转换成double类型时会丢失精度与实际结果相差0.0000000001

解决方法：将double换为float减少精度丢失

测试用例3：

15 10 4015

100 220 90 400 300 400 205 120 160 580 400 140 100 1300 650

8 24 13 80 70 80 45 15 28 90 130 32 20 120 40

8 44 13 100 100 90 75 25 28 120 130 32 40 160 40

3 6 4 20 20 30 8 3 12 14 40 6 3 20 5

5 9 6 40 30 40 16 5 18 24 60 16 11 30 25

5 11 7 50 40 40 19 7 18 29 70 21 17 30 25

5 11 7 55 40 40 21 9 18 29 70 21 17 35 25

0 0 1 10 4 10 0 6 0 6 32 3 0 70 10

3 4 5 20 14 20 6 12 10 18 42 9 12 100 20

3 6 9 30 29 20 12 12 10 30 42 18 18 110 20

3 8 9 35 29 20 16 15 10 30 42 20 18 120 20

550 700 130 240 280 310 110 205 260 275将以上数据存入txt文件中由Java程序文件流调用此txt文件实现测试数据输入

|  |  |
| --- | --- |
| 预期结果 | 实际结果 |
| 输出多维背包问题最大价值3800 | 和预期一致 |

缺陷和限制

由于运用的是精确算法，对于大规模输入计算机处理不过来，算法优化通过回溯法01背包问题改进。

建议

用于计算小规模输入多约束背包问题

评价

能基本解决小规模多约束背包问题

伪代码

import 引包

class 文件流类{

初始化变量

FileReader file；

读入txt文件放入初始化的变量中

}

class 主类{

初始化变量

boolean ok(double[] c\_shuxing, int m, double[] limite, int t,double[][] shuxing\_all){

for (i = 0; i < m; i++) {

if (第i个属性未越界){

return true；

else

return false；

}

void backtrack(int t){//回溯方法

if (是否到根节点) {

if (判断当前值是否大于最大值) {

bestv = c\_value;

}

}

else

搜索左子树

}

main方法{

实例化主类

调用主类方法

输出

}

}