# 2022秋-算法设计与分析

# 分支限界实验报告

|  |  |
| --- | --- |
| 姓名： |  |
| 学号： |  |

|  |
| --- |
| 2022年12月23日 |

目录

[2022秋-算法设计与分析 1](#_Toc124442408)

[分支限界实验报告 1](#_Toc124442409)

[1. 实验要求 2](#_Toc124442410)

[2. 实验报告 2](#_Toc124442411)

[2.1 回溯法和分支限界法解决整数背包问题 2](#_Toc124442412)

[**2.1.1 回溯法解决整数背包问题** 2](#_Toc124442413)

[**2.1.2 分支限界法解决整数背包问题** 3](#_Toc124442414)

[**2.1.3 运行结果** 4](#_Toc124442415)

[2.2 固定物品种类数量，产生大量随机样本 4](#_Toc124442416)

[2.3 蒙特卡洛法对分支数量估计 5](#_Toc124442417)

[**2.3.1 蒙特卡洛法实现** 5](#_Toc124442418)

[**2.3.2 蒙特卡洛法分支估计** 5](#_Toc124442419)

[**2.3.3 蒙特卡洛法应用建议** 6](#_Toc124442420)

# 1. 实验要求

对分支限界算法进行分析，理解算法的工作流程。具体要求如下：

1. 针对整数背包问题，实现回溯法和分支限界算法；

2. 以物品种类数n为输入规模，固定n，随机产生大量测试样本；

3. 用蒙特卡洛法对分支数量进行估计，根据统计结果给出应用蒙特卡洛估计整数背包问题分支数量的建议；

4. 在分支限界法计算代价函数时，使用其他算法（如动态规划法）计算出真实值，记录结点的代价函数与真实值，分析在同一输入规模下不同层代价函数的近似效果；

5. 改变n，分析在不同输入规模下同一层代价函数的近似效果；

6. 针对实验结果，设计新的代价函数，并分析其有效性。

# 2. 实验报告

## 2.1 回溯法和分支限界法解决整数背包问题

### **2.1.1 回溯法解决整数背包问题**

使用函数backtrace()实现回溯法解决整数背包问题。backtrace()函数接收四个参数，分别是记录背包最大承重maxWeight，物品价值values，物品重量weights，以及物品个数n。使用函数backtraceHelper()通过递归实现回溯法，递归中需记录当前搜索第i号物品，当前所有装入物品的总价值tempValue，当前所有装入物品的总重量tempWeight，以及目前为止可行解的最大价值maxValue。

整数背包问题中，每个物品的数量是无限个，因此，在backtraceHelper()函数中，分别选择0, 1, 2…个第i号物品，直到无法装入num个第i号物品。每次装入num个第i号物品后，更新tempValue和tempWeight，然后递归调用backtraceHelper()函数自身，传入第i+1号物品，直到i等于n，说明已经搜索了所有物品。通过比较maxValue和tempValue 的大小，选择是否更新maxValue。

while循环中，每次num值增加1，如果当前无法装入num个第i号物品，则退出循环，直接返回，不用继续搜索子结点。

具体实现如下图所示：

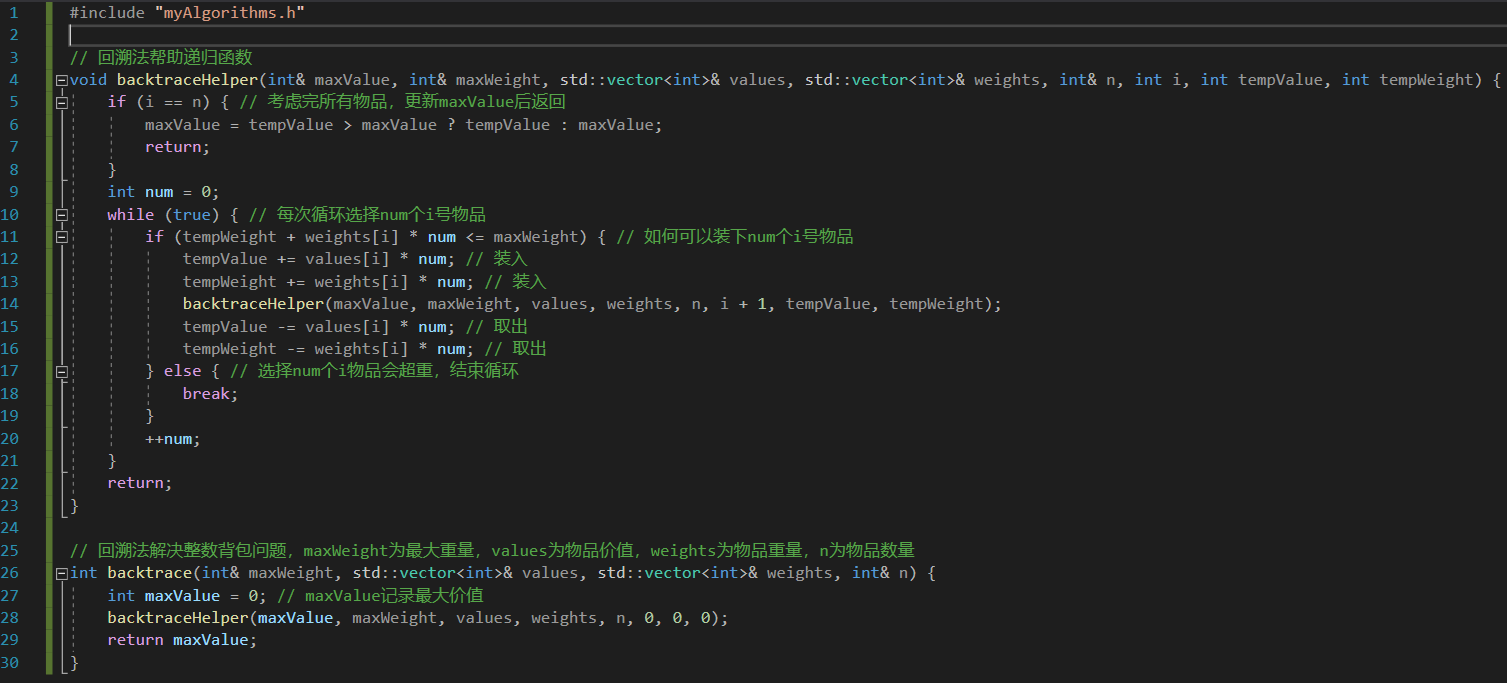


图1：回溯法解决整数背包问题的实现

### **2.1.2 分支限界法解决整数背包问题**

分支限界法解决整数背包问题，代价函数是当前已装入的物品总价值加上还可以继续装入的最大价值的上界，因此需要先根据物品性价比由大到小排序。branchAndBound()函数接收与backtrace()函数一样的参数，但首先对value和weight数组进行排序，使用选择排序，根据性价比进行排序，如下图所示：

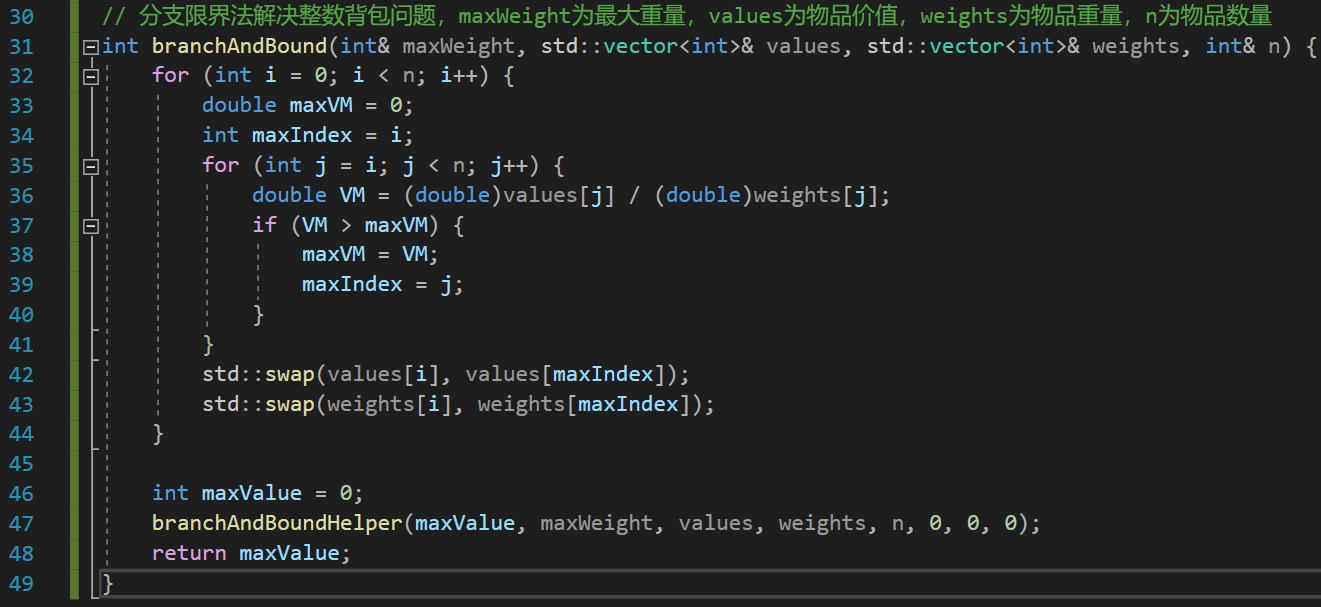


图2：branchAndBound()函数的实现

排序完成后，就可以将参数传入branchAndBoundHelper()函数使用递归实现回溯法，基本逻辑与backtraceHelper()一致，区别是在while循环之前，需要计算代价函数与界，即maxValue进行比较，如果代价函数小于界，则直接返回。具体实现使用potential表示代价函数，具体实现如下图所示：

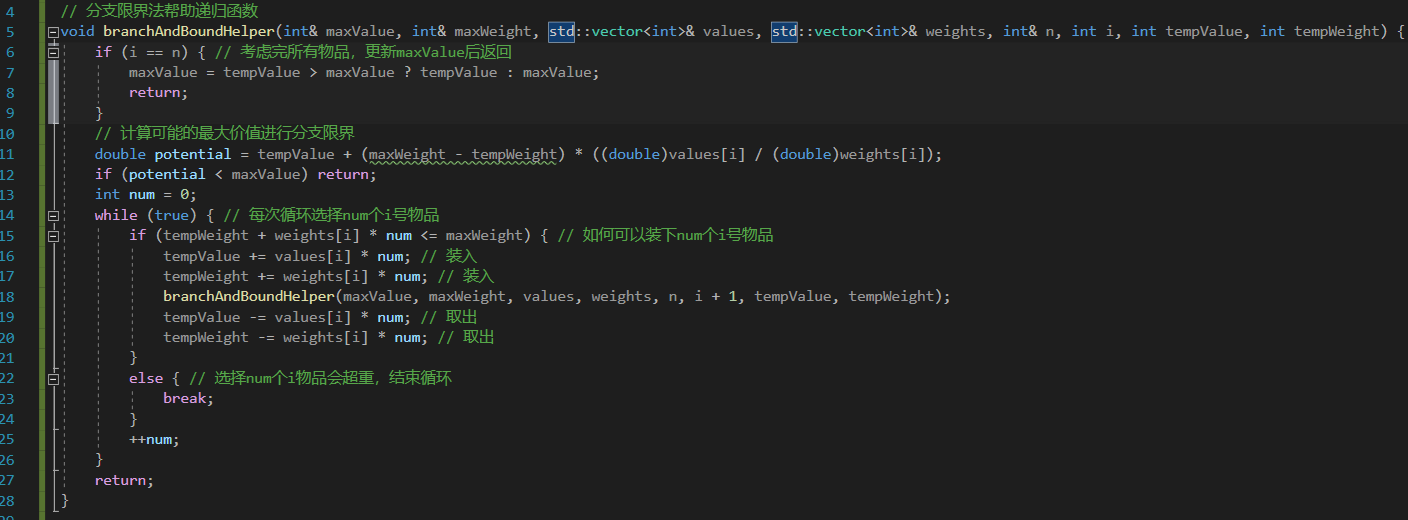


图3：branchAndBoundHelper()函数的实现

### **2.1.3 运行结果**

采用PPT中示例，values数组为[1, 3, 5, 9]，weights数组为[2, 3, 4, 7]。调用两个算法进行求解，结果如下图所示：

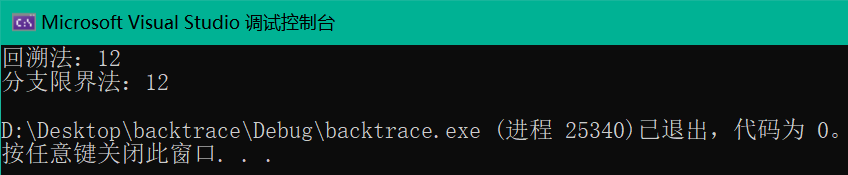


图4：运行结果

## 2.2 固定物品种类数量，产生大量随机样本

固定物品种类数量n不变，设置好背包总重量和每个物品的最大重量，然后设置好随机样本的数量。每个随机样本是n个物品，使用for循环，循环次数设置为随机样本数量，在每次循环内部，使用rand()函数生成n个物品的重量：



图5：产生随机样本

## 2.3 蒙特卡洛法对分支数量估计

### **2.3.1 蒙特卡洛法实现**

蒙特卡洛法使用大量随机过程模拟分支数量，每次算法，在搜索树扩展过程中，随机选择扩展节点，多次模拟搜索树扩展过程，记录扩展出的节点的平均数量，即可表示分支数量：

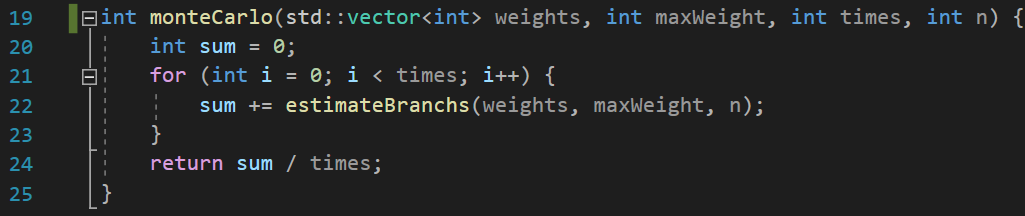


图6：蒙特卡洛法主函数

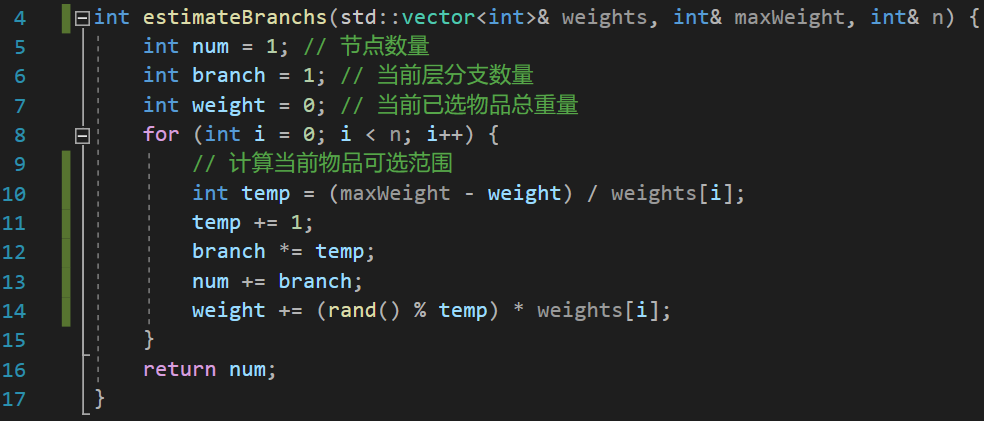


图7：蒙特卡洛法估计分支函数

### **2.3.2 蒙特卡洛法分支估计**

改变蒙特卡洛法内部循环次数，统计出蒙特卡洛法内部循环次数与估计出的分支数量的关系，如下表所示：

表1：蒙特卡洛法循环次数与分支数量关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 循环次数 | 样本1 | 样本2 | 样本3 | 样本4 |
| 10 | 14511 | 84891 | 28508 | 47835 |
| 100 | 47943 | 1038221 | 417754 | 248626 |
| 1000 | 53562 | 1392036 | 411001 | 230858 |
| 10000 | 490302 | 1189282 | 412846 | 237293 |

### **2.3.3 蒙特卡洛法应用建议**

从统计结果可以明显看出，使用蒙特卡洛法估计分支数量，随着内部循环次数的增加，统计结果趋于稳定。因此，使用蒙特卡洛法时，需要设置合理的循环次数，至少设置为每个测试样本中物品数量n的平方次，才能得到相对精确的结果。