算法设计与分析第三章作业

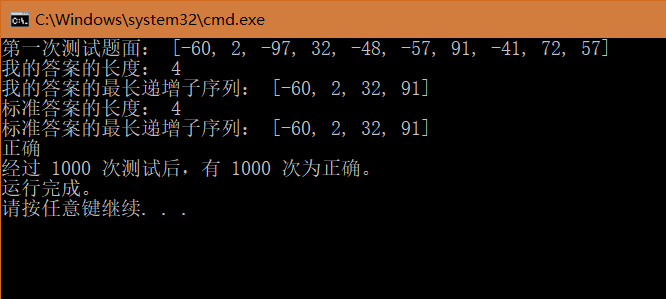
1652270 冯舜

# 算法分析题3-1

设序列用L[i]表示。用一个二位数组x[i]来表示L中以L[i]结尾的最长单调增子序列的长度。则所需求的即为

并且，x[i]对应的子序列的倒数第二个元素L[j]也对应一个x[j]，故该问题解具有最优子结构性质；很明显，子序列可能被之后的多个子序列重用，故问题解具有重叠子问题性质，以此进行动态规划：

具体过程参见3-1.py（可看注释、或用python 3.6运行）。运行结果：



随机生成了1000个数据以及用已有的算法生成的结果作为参照，全部验证通过。且时间复杂度。

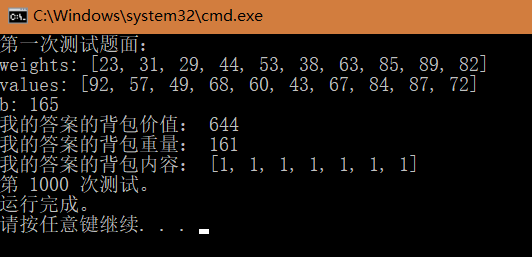
**更新：用复杂度解决问题**

维护一个数组X[i]，记录序列中长度为i+1的递增子序列中，最小的末位元素对应在L中的下标。用遍历下标i扫描序列L时，L[i]大于X[i]，便把i添到X[i+1]；否则将i二分插入X[i]中（）。这样操作后，记录L[i]作为递增子序列末位元素的前一个元素。扫描完后，追踪X的最后一个元素直至到没有前一个元素，追踪出来的序列即为最长的递增子序列。详情参照3-1.py的longestIncrementalSubsequence\_NlgN函数。

# 算法分析题3-3

此即一般性的背包问题，与课本中的0-1背包问题的解法唯一不同为引用“容量为j-wi”的子问题时不要求可选物品的种数减一。

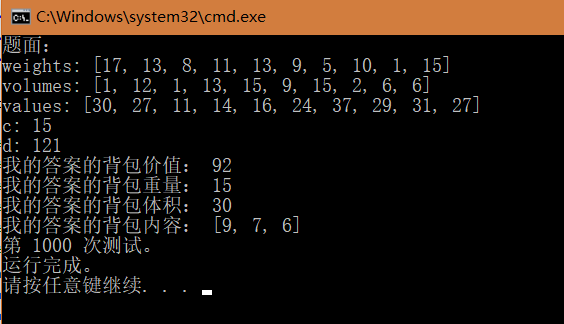
具体过程可看3-3.py，时间复杂度为。



如图，通过了1000次计算结果一致性测试。

# 算法分析题3-4

与0-1背包问题基本相同，只是多了容量限制，在开数组时要多开一个维度，且在递推关系中引用的子问题除了要使j减去wi也要使允许体积k减去vi（第i物品的体积）。详细过程见3-4.py，。

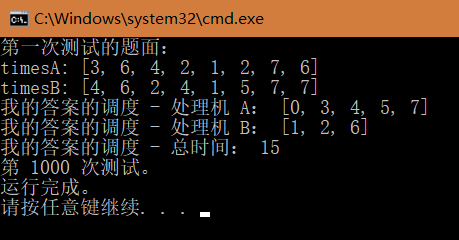


# 算法实现题3-1

需要用一个包括“限制A运行时间i”、“限制B运行时间j”、“限制作业的范围为从第1个到第k个”三个维度、内容为“能否完成任务的布尔值”的数组X来进行线性规划。

递推关系为：X[i][j][k] = X[i-ak][j][k-1] || X[i][j-bk][k-1]。

详细过程见3-i1.py。由于i的范围为，j的范围为，故时间复杂度为。



# 算法实现题3-3

若石堆为线性，线性规划的思想与课本中矩阵连乘的思想相同，都是就线性元素的划分将问题X[i][j]分割为子问题X[i][k], X[k+1][j]。实际上，石堆为环形也可利用这个方法，只是将第二个下标j重新定义为从i开始的步长，来确定原问题的范围；同时在处理环形的时候，利用取模运算让下标越过n后重新从零开始即可。计算数组完毕后，固定步长为n-1，遍历所有的起始范围i，确定最大分和最小分。

详细过程参看3-i3.py。时间复杂度为。

