算法设计与分析第六章作业

1652270 冯舜

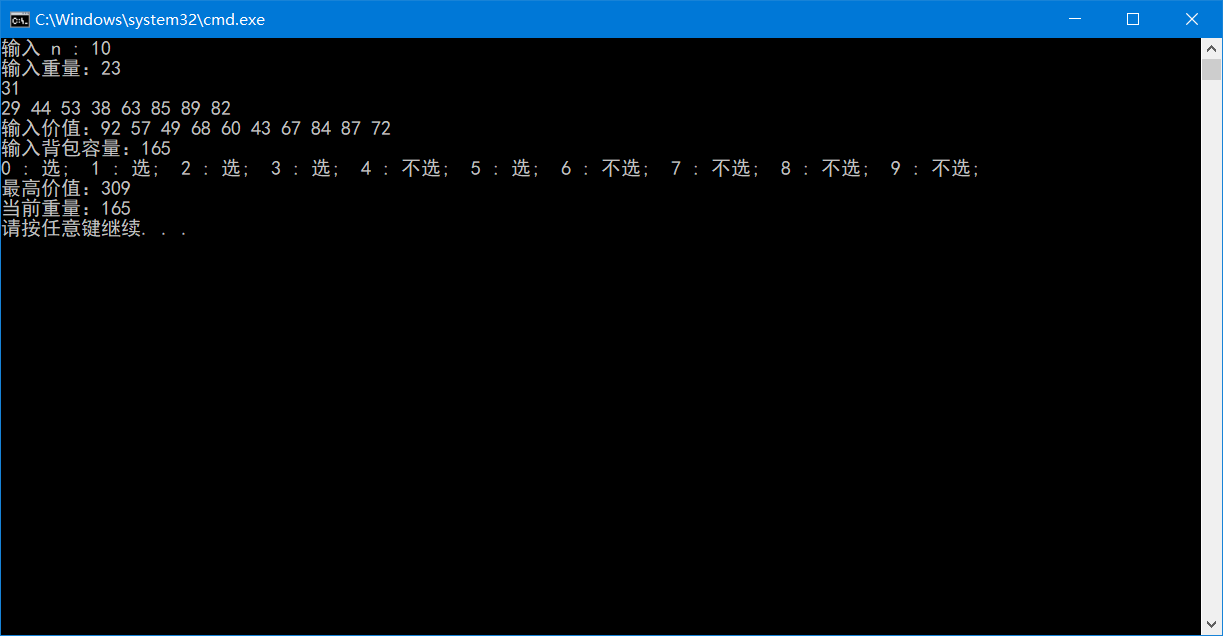
# 算法分析题6-1

该算法与课本中用优先队列式分支限界法解0-1背包问题的算法基本相同，不同点在于。数据结构上，课本中解法用队列，本算法用栈。

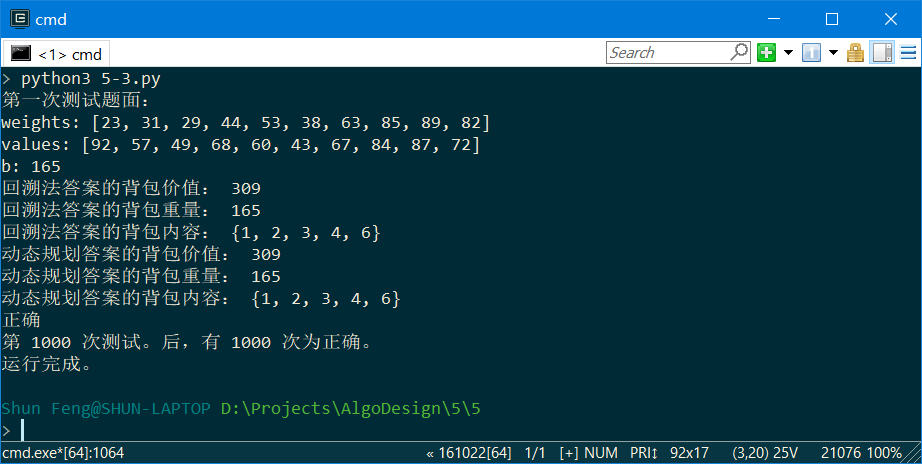
该算法与回溯法虽然都是以广度优先搜索的方式来遍历搜索树，但它们也有不同之处：

* 栈式分支限界法在处理完一个活节点后，这个节点即刻变为死节点而不会重新变回活节点；回溯法不同，在回溯到之前的节点后，曾经的活节点仍有可能重新成为活节点而被扩展。
* 栈式分支限界法一次生成活节点的所有子节点入栈，但回溯法不是如此，而是一个一个生成子节点，并递归遍历刚生成的子节点。后面的子节点有可能因为上一个子节点的遍历结果而被剪枝。

算法具体见工程中的 6-1.cpp。运行结果：



与之前用回溯法、动态规划法的解相比较，相同：

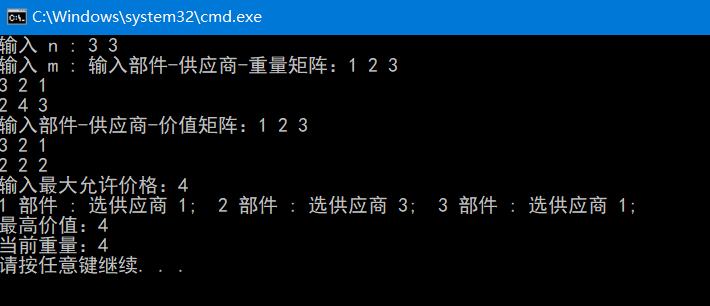


# 算法实现题6-4

用优先队列分支限界法搜索子集树，优先级用当前选择好的部件总重量来判定，越轻越优先。

具体算法见 6-i4.py。。

运行结果：（数据的重量第三行改为了2 4 3以突出差别）

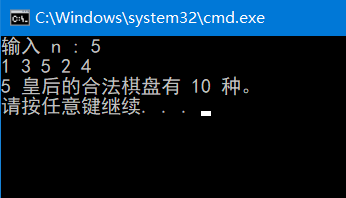


# 算法实现题6-6

用队列式分支限界法搜索存储每行应放置王后位置的八叉树，搜索到解后立即返回即可。

这里的算法并没有立即返回，意在统计王后放置的方法数。

具体算法实现见 6-i6.py。运行结果：



# 算法实现题6-9

用优先队列式分支限界法搜索排列树，以算式短为优先（结点越浅越优先），返回得数为m的节点对应的算式。只不过在一个新的数字确定之后，创建四个而不是一个子节点，代表四种运算符。

具体实现见6-i9.py。运行结果如下：

