**从VB到Python之赢取最多奖金**

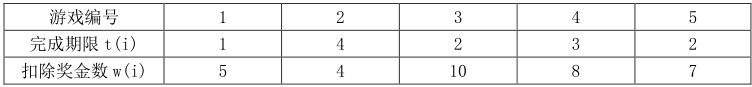
余姚二中 梁见斌

VB代码：

17. 【加试题】(2016年6月金华市期末十校联考)小炫报名参加“智力大冲浪”节目。比赛规则如下：

比赛开始时参赛者将预先得到 M 元奖金。

比赛时间分为 N 个时段(N≤100)，有 N 个小游戏，每个时段可完成 1 个游戏，第 i 个小游戏必须在规定期限 t(i)时段前完成(1≤t(i)≤N)，否则要从奖金 M 元中扣去奖金 w(i)，w(i)为自然数，不同的游戏扣去的奖金是不一样的。每个游戏必须从整时段开始。注意：比赛绝对不会让参赛者赔钱!

例如：当 N=5,M=100 时

在这种情况下，小炫依次完成编号为 5、3、4、2、1 的小游戏，获得了最高奖金 95 元。他的算法思想：让扣款高的游戏尽量准时完成。扣除的奖金越少，则最后赢取的奖金越多。

（1）按扣款数值从大到小排序，在样例中，排序后游戏编号依次为 3、4、5、1、2；

（2）对于游戏 i，在时间段 1 到 t(i)完成，效果都是一样的，所以尽量放置的时间段 t(i)完成，若该时段已经被占用，则依次考查时间段 t(i)-1, t(i)-2,...,1。在样例中，

1. 考虑游戏 3，放置在时间段 2 完成（注：t(3)=2）；

2. 考虑游戏 4，放置在时间段 3 完成（注：t(4)=3）；

3. 考虑游戏 5，时间段 2 已被游戏 3 占用，放置在时间段 1 完成（注：t(5)-1=1）；

4. 考虑游戏 1，时间段 1 已被游戏 5 占用，不能按时完成，记录扣款；

5. 考虑游戏 2，放置在时间段 4 完成（注：t(2)=4）

小炫按如上算法编写了一个 VB 程序，计算“智力大冲浪”游戏中玩家最多可赢取的奖金，将结果输出到文本框 Text1 中。

VB 程序代码如下所示，请在划线处填入合适代码。

Dim n As Integer, m As Integer

Dim t(1 To 100) As Integer '变量 t(i) 表示第 i 个游戏的完成期限

Dim w(1 To 100) As Integer '变量 w(i) 表示第 i 个游戏未完成要扣除的奖金

Dim f(1 To 100) As Boolean '变量 f(i)表示第 i 个时段是否已经被占用

Private Sub Form\_Load()

'初始化游戏设置信息，时段数量 n、初始奖金数 m 、数组 t、数组 w

'代码略

End Sub

Sub swap(x As Integer, y As Integer) '自定义过程，可以用 Call 语句来调用该过程

Dim z As Integer

z = x: x = y: y = z

End Sub

Sub sort() '自定义过程，可以用 Call 语句来调用该过程

Dim x As Integer, i As Integer, j As Integer

For i = 1 To n - 1

For j = i + 1 To n

If w(i) < w(j) Then

Call swap(w(i), w(j)) '调用自定义过程

①

End If

Next j

Next i

End Sub

Private Sub Command1\_Click()

Dim i As Integer, k As Integer, p As Integer

Call sort '调用自定义过程

tot = 0

For i = 1 To n

f(i) = True

Next i

For i = 1 To n '对每个游戏从该游戏的规定期限开始往左找可用时间段

'找到可用时间段则完成该游戏，否则放弃该游戏

p = -1

k = t(i)

Do While k > 0 And p = -1

If f(k) = True Then

p = k

f(k) = False

Else

②

End If

Loop

If p = -1 Then tot = tot + w(i)

Next i

③

Text1.Text = Val(ans)

End Sub

答案：（1）call swap(t(i),t(j)) (2分)

（2）k = k - 1 (2分)

（3）ans = m - tot (2分)

算法分析：因为不同的小游戏不能准时完成时具有不同的扣款权数，而且是最优解问题，所以本题很容易就想到了贪心法。贪心的主要思想是要让扣款数值大的尽量准时完成。这样我们就先把这些任务按照扣款的数目进行排序，把大的排在前面，先进行放置。

假如罚款最多的一个任务的完成期限是k，我们应该把它安排在哪个时段完成呢？应该放在第k个时段，因为放在1～k任意一个位置，效果都是一样的。

若第k个时段已经有任务占据，则向前查询是否有空的时间段，若完成期限之前的时间段全部被占据，则放弃该任务。这样能获得最优结果。

题目提供的示例，充满说明了本算法思想。

对应Python代码：

def fun(a): #返回扣除奖金的最小值

b = [True] \* len(a)

c = 0

for i in range(len(a)):

k = a[i][1]

while k > 0 and not b[k]:

k -= 1

if k > 0:

b[k] = False

else:

c += a[i][0]

return c

with open('zldcl.txt', 'r') as fin:

m = int(fin.readline()) #预发m元奖金

n = int(fin.readline()) #总共n个学生

t = tuple(map(int, fin.readline().split()))#t(i)指第i个游戏的完成期限

w = tuple(map(int, fin.readline().split()))#w(i)指未完成第i个游戏扣除的奖金

a = list(zip(w, t))

a.sort(key=lambda x: x[0], reverse=True)

c = m - fun(a)

print(c)

从上例可以看出，利用Python语言丰富的内置函数和强大的高阶函数，可以很方便的解决对列表的排序问题，只需少量代码就顺利地表达了算法思想，直指问题的核心。