# Assignment #5: Greedy穷举Implementation

Updated 1939 GMT+8 Oct 21, 2024

2024 fall, Complied by <mark>祁黄奕 城市与环境学院</mark>

\*\*说明：\*\*

1）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3）提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

4）如果不能在截止前提交作业，请写明原因。

## 1. 题目

### 04148: 生理周期

brute force, http://cs101.openjudge.cn/practice/04148

思路：在给出的日子到这个日子与21253的和之间，找到一个数对各个周期数取模的值，与各个周期数达到peak的数对其对应周期数取模的值相等即可。

代码：

```python

def find\_next\_peak(p, e, i, d):  
 cycle\_p = 23  
 cycle\_e = 28  
 cycle\_i = 33  
 x = 0  
 for t in range(d + 1, d+21253):  
 if t % cycle\_p == p % cycle\_p and t % cycle\_e == e % cycle\_e and t % cycle\_i == i % cycle\_i:  
 x = t  
 break  
 return x - d  
  
num=1  
while True:  
 p, e, i, d = map(int, input().split())  
 if p == e == i == d == -1:  
 break  
 print(f"Case {num}: the next triple peak occurs in {find\_next\_peak(p, e, i, d)} days.")  
 num+=1

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 18211: 军备竞赛

greedy, two pointers, http://cs101.openjudge.cn/practice/18211

思路：运用双指针，从从小到大排好序的设计图金额列表左右同时查找，若最小值<本金，则本金-最小值，我方武器数量+1，列表最左边的值无需再考虑。在保证我方武器数大于对方的情况下，可售出最贵武器，本金金额加上武器金额。

代码：

```python

p = int(input())  
w = list(map(int, input().split()))  
w.sort()  
a = 0  
b = 0  
left = 0  
right = len(w) - 1  
  
while left <= right:  
 if p >= w[left]:  
 p -= w[left]  
 a += 1  
 left += 1  
 elif a > b and left < right:  
 p += w[right]  
 b += 1  
 right -= 1  
 else:  
 break  
print(a - b)

```

代码运行截图 ==（至少包含有"Accepted"）==

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 21554: 排队做实验

greedy, http://cs101.openjudge.cn/practice/21554

思路：将输入的列表建立其原值与原索引对应的列表，分别排序，建立仅含值排序后所对应的原索引值列表，将其输出，并输出各个时间累加和序列除以总人数的结果即可。

代码：

```python

n=int(input())  
a=list(map(int,input().split()))  
index=[(value,index) for index,value in enumerate(a,start=1)]  
a=sorted(a)  
b=sorted(index)  
new=[index for value,index in b]  
new=' '.join(map(str,new))  
print(new)  
c=0  
lst=[]  
for i in range(len(a)-1):  
 c+=a[i]  
 lst.append(c)  
print("%.2f"%(sum(lst)/n))

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

图形用户界面, 文本, 应用程序

描述已自动生成

### 01008: Maya Calendar

implementation, http://cs101.openjudge.cn/practice/01008/

思路：建立haab月份与编号对应的字典、tzolkin名称的列表，将输入的haab日期转为总天数（年数\*365 +月份天数+日数），将总天数转为tzolkin日期（数字 = (总天数 % 13) + 1，名字 = 总天数 % 20，年份 = 总天数 // 260）。对每个哈布历日期调用函数得到tzolkin日期。输出日期数量 n及各tzolkin日期

代码：

```python

# 定义哈布历的月份和天数  
haab\_months = {  
 'pop': 0, 'no': 1, 'zip': 2, 'zotz': 3, 'tzec': 4,  
 'xul': 5, 'yoxkin': 6, 'mol': 7, 'chen': 8, 'yax': 9,  
 'zac': 10, 'ceh': 11, 'mac': 12, 'kankin': 13, 'muan': 14,  
 'pax': 15, 'koyab': 16, 'cumhu': 17, 'uayet': 18  
}  
  
tzolkin\_names = [  
 'imix', 'ik', 'akbal', 'kan', 'chicchan', 'cimi', 'manik', 'lamat', 'muluk', 'ok',  
 'chuen', 'eb', 'ben', 'ix', 'mem', 'cib', 'caban', 'eznab', 'canac', 'ahau'  
]  
  
n = int(input())  
dates = [input().strip() for \_ in range(n)]  
  
def haab\_to\_days(date):  
 day, month, year = date.split()  
 day = int(day[:-1])  
 month = haab\_months[month]  
 year = int(year)  
  
 total\_days = year \* 365 + (month \* 20 + day if month < 18 else 18 \* 20 + day)  
 return total\_days  
  
  
def days\_to\_tzolkin(days):  
 number = (days % 13) + 1  
 name = tzolkin\_names[days % 20]  
 return f"{number} {name} {days // 260}"  
  
results = [days\_to\_tzolkin(haab\_to\_days(date)) for date in dates]  
  
print(n)  
for result in results:  
 print(result)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



### 545C. Woodcutters

dp, greedy, 1500, https://codeforces.com/problemset/problem/545/C

思路：建立树的坐标和高度列表，最后一棵树肯定能砍所以算一，遍历列表中的值，能往左或往右就加一。Count标记已倒下的树的最右端，初始时让第一颗树一定能往左倒。

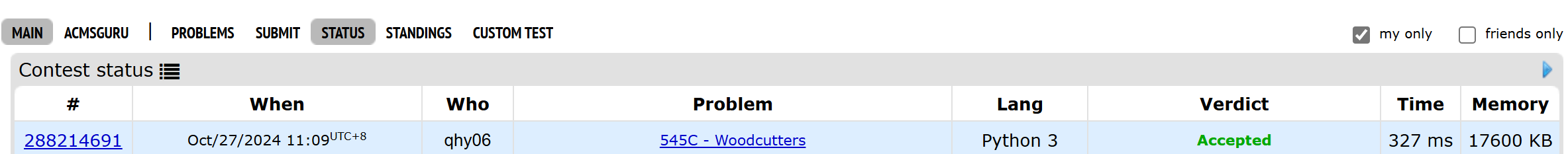
代码：

```python

n = int(input())  
trees = [[int(a) for a in input().split()] for \_ in range(n)]  
count = 1  
r = -trees[0][1]  
for i in range(n-1):  
 if trees[i][0]-trees[i][1] > r:  
 count += 1  
 r = trees[i][0]  
 elif trees[i][0]+trees[i][1] < trees[i+1][0]:  
 count += 1  
 r = trees[i][0]+trees[i][1]  
 else:  
 r = trees[i][0]  
print(count)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



### 01328: Radar Installation

greedy, http://cs101.openjudge.cn/practice/01328/

思路：将所有岛屿的水平覆盖范围转化为区间并从小到大排序，然后找出这些区间中不重叠的区间数量。

代码：

```python

from math import sqrt  
case\_cnt = 0  
while True:  
 n, d = map(int, input().split())  
 if not n and not d:  
 break  
 case\_cnt += 1  
 flag = True  
 isl = [list(map(int, input().split())) for \_ in range(n)]  
 for i in range(n):  
 if isl[i][1] > d:  
 flag = False; break  
 isl[i] = [isl[i][0] - sqrt(d \*\* 2 - isl[i][1] \*\* 2), isl[i][0] + sqrt(d \*\* 2 - isl[i][1] \*\* 2)]  
 if not flag:  
 print(f"Case {case\_cnt}: -1")  
 input(); continue  
 isl = sorted(isl, key=lambda x: x[1])  
 cur, ans = isl[0][1], 1  
 for i in range(n):  
 if isl[i][0] > cur:  
 cur = isl[i][1]; ans += 1  
 print(f"Case {case\_cnt}: {ans}")  
 input()

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

图形用户界面, 文本, 应用程序

描述已自动生成

## 2. 学习总结和收获

作业好难啊真的好难啊！做这次的作业用了一天，感觉自己的能力水平与课程难度不匹配，虽然看解释最后基本能理解，但还是被狠狠重击了。Dp还没学懂，认识了双指针，对greedy还是一知半解，算法的概念好复杂。一直在以蜗牛爬行的速度学习着，试图记住更多函数。正在极其缓慢地跟进每日选做中……