

Assignment #5: Greedy 穷举 Implementation

Updated 1939 GMT+8 Oct 21, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</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：

因为数值不是太大采用计算机思维，选定 a 为标准不断加它的周期，然后看看每一个 a 与初始的 b 和 c 的差值是不是 b 和 c 的各自周期的整数倍，如果是就说明这一天不仅 a 的高峰可以到， b 和 c 在经过若干个周期后也是可以到的。这样的好处是不用考虑 b 和 c 的变化，而仅仅考虑 a 的变化带来的与 b 和 c 的差值变化。

代码：

```
i=1
while True:
    a,b,c,d=map(int,input().split())
    if {a,b,c,d}=={-1}:
        break
    flag=True
    while flag:
        a+=23
        if (a-b)%28==0 and (a-c)%33==0:
            flag=False
    print(f'Case {i}: the next triple peak occurs in {a-d} days.')
    i+=1
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



#46375243提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
i=1
while True:
    a,b,c,d=map(int,input().split())
    if {a,b,c,d}=={-1}:
        break
    flag=True
    while flag:
        a+=23
        if (a-b)%28==0 and (a-c)%33==0:
            flag=False
    print(f'Case {i}: the next triple peak occurs in {a-d} days.')
    i+=1
```

基本信息

#: 46375243
题目: 04148
提交人: 24n2400016635
内存: 3596kB
时间: 19ms
语言: Python3
提交时间: 2024-10-09 14:38:42

18211: 军备竞赛

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

思路:

使用双指针从排序后的价格的两端向中间靠拢，钱够就从左边制作，钱不够就从右边卖，这样制作花费少，赚的钱还多，就非常 greedy 了。同时在 while 循环要保证制作的要大于等于卖出去的 ($i \geq n-j-1$)，最后 i 和 j 相等时再额外判断一下此时能不能买。

代码:

```
p=int(input())
s=sorted(list(map(int,input().split())))
n=len(s)
i,j=0,n-1
if p>=s[0]:
    while (i>=n-j-1) and i<=j:
        if p>=s[i]:
            p-=s[i]
            i+=1
        else:
            p+=s[j]
            j-=1
        if i==j:
            if p>=s[i]:
                i+=1
            else:
                break
    print(i-n+j+1)
else:
    print(0)
```

代码运行截图 == (至少包含有"Accepted") ==



#46484357提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
p=int(input())
s=sorted(list(map(int,input().split())))
n=len(s)
i,j=0,n-1
if p>=s[0]:
    while (i>=n-j-1) and i<=j:
        if p>=s[i]:
            p-=s[i]
            i+=1
        else:
            p+=s[j]
            j-=1
        if i==j:
            if p>=s[i]:
                i+=1
            else:
                break
    print(i-n+j+1)
else:
    print(0)
```

基本信息

#: 46484357

题目: 18211

提交人: 24n2400016635

内存: 3640kB

时间: 20ms

语言: Python3

提交时间: 2024-10-14 18:48:39

21554: 排队做实验

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

思路:

很显然是按照时间排序, 时间短的先做实验。但是有一个陷阱是如果用字典储存,

```
for index,value in enumerate(t):
    adict[value]=index+1
```

假设存在两个相同的时间, 那么第一个时间的 `index` 将会被更新成第二个时间的 `index` (即两个时间的 `index` 相同了), 这样不符合题意。所以采取列表储存一对数值。

代码:

```
n=int(input())
t=list(map(int,input().split()))
d=[]
for index,value in enumerate(t):
    d.append([index+1,value])
d.sort(key=lambda x:x[1])
index=[]
ans=0
for i in range(0,n):
    index.append(d[i][0])
    ans+=d[i][1]*(n-i-1)
print(*index)
print(format(ans/n, '.2f'))
```

代码运行截图 `<mark>` (至少包含有"Accepted") `</mark>`

状态: Accepted

源代码

```
n=int(input())
t=list(map(int,input().split()))
d=[]
for index,value in enumerate(t):
    d.append([index+1,value])
d.sort(key=lambda x:x[1])
index=[]
ans=0
for i in range(0,n):
    index.append(d[i][0])
    ans+=d[i][1]*(n-i-1)
print(*index)
print(format(ans/n,'.2f'))
```

基本信息

#: 46648625
题目: 21554
提交人: 24n2400016635
内存: 3636kB
时间: 21ms
语言: Python3
提交时间: 2024-10-21 23:36:51

01008: Maya Calendar

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

思路:

本能反应还是打字典，没想到答案里面的列表，列表还是比较方便。先计算原来的总天数，然后转化时有一个坑，当天数恰好是 260 倍数的时候，转化后的日期是 days//260-1 年份的最后一个月的最后一天。（举例：假如恰好 260 天，那么输出应该是 13 ahau 0 而不是 1 imix 1，可以想成现实生活的第 365 天是一年的最后一天，而不是下一年的第一天）

最痛苦的是 debug 的过程，一开始没想到这个陷阱，看了群才知道。

刚开始把题目的单词复制错了，找半天错误才发现漏了一个字母。。

代码:

```
Haab_month={
    'pop':1, 'no':2, 'zip':3, 'zotz':4, 'tzec':5,
    'xul':6, 'yoxkin':7, 'mol':8, 'chen':9, 'yax':10,
    'zac':11, 'ceh':12, 'mac':13, 'kankin':14, 'muan':15,
    'pax':16, 'koyab':17, 'cumhu':18, 'uayet':19
}
Tzolkin={1:'imix', 2:'ik', 3:'akbal', 4:'kan',
          5:'chicchan', 6:'cimi', 7:'manik',
          8:'lamat', 9:'muluk', 10:'ok', 11:'chuen',
          12:'eb', 13:'ben', 14:'ix', 15:'mem', 16:'cib',
          17:'caban', 18:'eznab', 19:'canac', 20:'ahau'}
n=int(input())
ans=[]
for _ in range(n):
    a,bc=input().split(' ')
    b,c=bc.split(' ')
    days=365*int(c)+20*(Haab_month[b]-1)+int(a)+1
    z=days//260
    rest=days%260
    #如果恰好是 260 的倍数，直接输出 z-1 和最后一月的最后一天
    if rest==0:
```

```

        ans.append([13, 'ahau', z-1])
    else:
        x,y=(rest-1)%13+1,Tzolkin[(rest-1)%20+1]
        ans.append([x,y,z])

print(n)
for i in ans:
    print(*i)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```

Haab_month={
    'pop':1, 'no':2, 'zip':3, 'zotz':4, 'tzec':5,
    'xul':6, 'yoxkin':7, 'mol':8, 'chen':9, 'yax':10,
    'zac':11, 'ceh':12, 'mac':13, 'kankin':14, 'muan':15,
    'pax':16, 'koyab':17, 'cumhu':18, 'uayet':19
}

Tzolkin=[1:'imix', 2:'ik', 3:'akbal', 4:'kan',
          5:'chicchan', 6:'cimi', 7:'manik',
          8:'lamat', 9:'muluk', 10:'ok', 11:'chuen',
          12:'eb', 13:'ben', 14:'ix', 15:'mem', 16:'cib',
          17:'caban', 18:'eznab', 19:'canac', 20:'ahau']

n=int(input())
ans=[]
for _ in range(n):
    a,bc=input().split(' ')
    b,c=bc.split(' ')
    days=365*int(c)+20*(Haab_month[b]-1)+int(a)+1
    z=days//260
    rest=days%260
    if rest==0:
        ans.append([13, 'ahau', z-1])
    else:
        x,y=(rest-1)%13+1,Tzolkin[(rest-1)%20+1]
        ans.append([x,y,z])

print(n)
for i in ans:
    print(*i)

```

基本信息

#: 46646732
 题目: 01008
 提交人: 24n2400016635
 内存: 3788kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-10-21 21:44:57

545C. Woodcutters

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

思路: 感觉这个 greedy 好像只有一直思路而且也是大家都能想到的, 对于一棵树先看能不能放到左边, 要是不能再看能不能放到右边, 以此类推。中间变量 mark 用来标记当前已经处理过的数的右边界, 防止向左放时重合。但是有一个坑是只有一棵树的时候答案是 1。

代码:

```

n=int(input())
s=[]
cnt=2
for _ in range(n):
    x,h=map(int,input().split())
    s.append([x,h])
if n==1:

```

```

    print(1)
else:
    mark=s[0][0]
    for i in range(1,n-1):
        if s[i][0]-mark>s[i][1]:
            cnt+=1
            mark=s[i][0]
        elif s[i+1][0]-s[i][0]>s[i][1]:
            cnt+=1
            mark=s[i][0]+s[i][1]
        else:
            mark=s[i][0]
    print(cnt)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
287232654	Practice: twj_ink	S45C - 14	Python 3	Accepted	343 ms	16348 KB	2024-10-21 18:02:06	2024-10-21 18:02:06	☆	Compare

→ Source Copy

```

n=int(input())
s=[]
cnt=2
for _ in range(n):
    x,h=map(int,input().split())
    s.append([x,h])
#如果只有1棵树显然只能fall一个
if n==1:
    print(1)
else:
    mark=s[0][0]
    for i in range(1,n-1):
        #与前一棵树之间有空位
        if s[i][0]-mark>s[i][1]:
            cnt+=1
            mark=s[i][0]
        #如果不行看与后一棵树的空位
        elif s[i+1][0]-s[i][0]>s[i][1]:
            cnt+=1
            mark=s[i][0]+s[i][1]
        #如果都不能要把mark更新到当前的树
        else:
            mark=s[i][0]
    print(cnt)

```

[Click to see test details](#)

01328: Radar Installation

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

思路：

这道题用了 3 天时间。刚开始做这道题时，确实能想到“进程检测”和“Saruman's Army”这两道贪心，都是从左往右把标记放在尽量靠右的地方，但是我一开始想到的是，按照排队的 $used=[0]*n$ ，然后从左往右取点，每取一个点就以这个点为圆的靠左边界，向右找最远的圆心，然后把这个确定的圆里面的岛屿全部标记成 1。这个是“Saruman's Army”的思路，但是是二维。

但是实现之后发现一直 WA，而且真的是看不出来哪里有问题（而且代码及其复杂，搞了 40 多行），然后上课听到老师的提示之后，才知道原来完全可以把每一个岛屿的雷达范围投影到 x 轴上，然后这就和“进程检测”几乎差不多了，之后代码就很好实现了。也就是这一个过程将二维的贪心过程变成了一维，从而更好实现。

代码：

```

from math import sqrt
turn=0

```

```

while True:
    turn+=1
    n,d=map(int,input().split())
    can_cover,f=[],False
    if {n,d}=={0}:
        break
    for _ in range(n):
        x,y=map(int,input().split())
        if y>d:
            f=True
        else:
            right=x+sqrt(d**2-y**2)
            left=x-sqrt(d**2-y**2)
            can_cover.append([left,right])
    if f==True:
        print(f'Case {turn}: -1')
        input()
        continue
    can_cover.sort(key=lambda x:x[1])
    cnt,mark=0,-float('inf')
    for coverage in can_cover:
        if coverage[0]>mark:
            cnt+=1
            mark=coverage[1]
    print(f'Case {turn}: {cnt}')
    input()

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#46659076提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from math import sqrt
turn=0
while True:
    turn+=1
    n,d=map(int,input().split())
    can_cover,f=[],False
    if {n,d}=={0}:
        break
    for _ in range(n):
        x,y=map(int,input().split())
        if y>d:
            f=True
        else:
            right=x+sqrt(d**2-y**2)
            left=x-sqrt(d**2-y**2)
            can_cover.append([left,right])
    if f==True:
        print(f'Case {turn}: -1')
        input()
        continue
    can_cover.sort(key=lambda x:x[1])
    cnt,mark=0,-float('inf')
    for coverage in can_cover:
        if coverage[0]>mark:
            cnt+=1
            mark=coverage[1]
    print(f'Case {turn}: {cnt}')
    input()

```

基本信息

#: 46659076
题目: 01328
提交人: 24n2400016635
内存: 3784kB
时间: 53ms
语言: Python3
提交时间: 2024-10-22 15:45:50

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

2. 学习总结和收获

<mark>如果作业题目简单，有否额外练习题目，比如：OJ“计概 2024fall 每日选做”、CF、LeetCode、洛谷等网站题目。</mark>

这次作业感觉排队做实验和砍树还算正常，生理周期，军备竞赛，雷达都是想到思路了就好做但是考试的时候感觉是想不到思路，而且想出来了也要痛苦地 debug 半天；日历转化的边界条件（恰好 260 天是一年的最后一天而不是下一年的第一天）不好想，也要痛苦地 debug 半天。

每日选做现在还有点差距，因为有些题卡几天都没有什么好的思路，算法题目能力提高还是要继续做题。在 cf 做了一些最近的比赛题，发现还是只能做 A 和 B 的题目，C 级题里面几乎都是 dp 和 greedy 等的结合，做不来。