

Assignment #7: 贪心和DP

Updated 0919 GMT+8 Oct 24, 2023

2023 fall, Compiled by 钟明衡 物理学院

说明:

1) 请把每个题目解题思路(可选), 源码Python, 或者C++/C (已经在Codeforces/Openjudge上AC), 截图(包含Accepted, 学号), 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、作业评论有md或者doc。

4) 如果不能在截止前提交作业, 请写明原因。

另外, CF的题目, 在洛谷有中文翻译, 例如 <https://www.luogu.com.cn/problem/CF1764C>

编程环境

操作系统: Windows_NT x64 10.0.19045

Python编程环境: Visual Studio Code 1.76.1

C/C++编程环境: Visual Studio Code 1.76.1

1. 必做题目

158B. Taxi

*special problem, greedy, implementation, 1100

<https://codeforces.com/problemset/problem/158/B>

思路:

输入时计算1、2、3、4个同学的小组各有几个

4人小组, 一个组一辆车; 3人小组可以和1人的拼车; 2人小组, 两个组一辆车, 最后可能剩下0或1个组, 可以再加入1的小组

先把4、3、2装好, 剩下的1先插空, 再另外装车

代码

```
1  n = int(input())
2  l = input().split()
3  a = 0
4  b = 0
5  c = 0
6  d = 0
7  ans = 0
8  for i in l:
9      if i == '1':
10         a += 1
11     elif i == '2':
12         b += 1
13     elif i == '3':
14         c += 1
15     elif i == '4':
16         d += 1
17 ans += d+int(b/2)+c
18 b = b % 2
19 if c >= a:
20     a = 0
21 else:
22     a -= c
23 ans += int(a/4)
24 a = a % 4
25 if a+2*b > 0 and a+2*b <= 4:
26     ans += 1
27 elif a+2*b > 4:
28     ans += 2
29 print(ans)
30
```

代码运行截图

General									
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
220986629	Practice: MinghengZhong	158B - 10	Python 3	Accepted	122 ms	2488 KB	2023-08-29 17:58:25	2023-08-29 17:58:25	★ Compare

→ Source
Copy

```

n = int(input())
l = input().split()
a = 0
b = 0
c = 0
d = 0
ans = 0
for i in l:
    if i == '1':
        a += 1
    elif i == '2':
        b += 1
    elif i == '3':
        c += 1
    elif i == '4':
        d += 1
ans += d + int(b/2) + c
b = b % 2
if c >= a:
    a = 0
else:
    a = c
ans += int(a/4)
a = a % 4
if a+2*b > 0 and a+2*b <= 4:
    ans += 1
elif a+2*b > 4:
    ans += 2
print(ans)

```

545D. Queue

greedy, implementation, sortings, 1300

<https://codeforces.com/problemset/problem/545/D>

思路：

首先把输入的时间从小到大排序

然后，从最小的开始，如果前面的总时间不大于这个服务时间，就使答案+1，且时间加上服务时间；如果当前的总时间大于了这个服务时间，则直接跳过，因为最优情况下，无论如何这个人也必定是失望的，不如直接把他拎到队伍最后面，即直接跳过

代码

```

1  n = int(input())
2  l = list(map(int, input().split()))
3  l.sort()
4  time = 0
5  ans = 0
6  for i in range(n):
7      if time <= l[i]:
8          ans += 1
9          time += l[i]
10 print(ans)
11

```

代码运行截图

General									
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
229522091	Practice: MinghengZhong	545D - 9	Python 3	Accepted	93 ms	13308 KB	2023-10-24 10:21:51	2023-10-24 10:21:51	★ Compare

→ Source
Copy

```

n = int(input())
l = list(map(int, input().split()))
l.sort()
time = 0
ans = 0
for i in range(n):
    if time <= l[i]:
        ans += 1
        time += l[i]
print(ans)

```

803A. Maximal Binary Matrix

constructive algorithms, 1400

<https://codeforces.com/problemset/problem/803/A>

思路：

首先，如果 $k > n^2$ ，不可能填完，输出-1

若 $k \leq n^2$ ，创建一个 $n \times n$ 的全为'0'的矩阵，直接按照字典序，在原本为'0'的位置 (i, j) 和 (j, i) 填'1'，同时 k 相应减去1或2

特别地，当 k 为1时，只允许在对角位置上填'1'

最后按顺序输出即可

代码

```

1  n, k = map(int, input().split())
2  if k > n*n:
3      print(-1)
4  else:
5      M = [['0']*n for _ in range(n)]
6      for i in range(n):
7          for j in range(n):
8              if M[i][j] == '0':
9                  if k > 1:
10                     if i == j:
11                         k -= 1
12                         M[i][j] = '1'
13                     else:
14                         k -= 2
15                         M[i][j] = '1'
16                         M[j][i] = '1'
17                 elif k == 1:
18                     if i == j:
19                         k -= 1
20                         M[i][j] = '1'
21                     break
22             else:

```

```

23         break
24     if k == 0:
25         break
26     for i in range(n):
27         print(' '.join(M[i]))
28

```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
229524376	Practice: MinghengZhong	803A - 23	Python 3	Accepted	46 ms	0 KB	2023-10-24 10:42:47	2023-10-24 10:42:47	☆	Compare

→ Source
Copy

```

n, k = map(int, input().split())
if k > n*n:
    print(-1)
else:
    M = [['0']*n for _ in range(n)]
    for i in range(n):
        for j in range(n):
            if M[i][j] == '0':
                if k > 1:
                    if i == j:
                        k -= 1
                        M[i][j] = '1'
                    else:
                        k -= 2
                        M[i][j] = '1'
                        M[j][i] = '1'
                elif k == 1:
                    if i == j:
                        k -= 1
                        M[i][j] = '1'
                    else:
                        break
                else:
                    break
            if k == 0:
                break
    for i in range(n):
        print(' '.join(M[i]))

```

1793C. Dora and Search

constructive algorithms, data structures, two pointers, 1200,

<https://codeforces.com/problemset/problem/1793/C>

思路:

使用双指针 l 和 r ，每次判断左右的值是否为最大或最小，如果都不是，此时的 l 和 r 就是结果

否则，左为max/min则 $l + 1$ ；右为max/min则 $r - 1$ ，且变更max/min的值，继续判断

为了加快判断的速度，一开始先把输入的数字按从小到大排序，也使用两头的指针 s 和 b ，每次左右出现max/min的时候，就 $s + 1$ 或 $b - 1$

如果 $r = l$ ，说明不能找到，输出-1

代码

```

1 t = int(input())
2 for _ in range(t):
3     n = int(input())
4     big = list(map(int, input().split()))
5     num = [0]*big


```

```

6      big.sort()
7      l = 1
8      r = n
9      s = 0
10     b = n-1
11     while True:
12         if r == l:
13             break
14         else:
15             if num[r] == big[s]:
16                 s += 1
17                 r -= 1
18             elif num[r] == big[b]:
19                 b -= 1
20                 r -= 1
21             elif num[l] == big[s]:
22                 s += 1
23                 l += 1
24             elif num[l] == big[b]:
25                 b -= 1
26                 l += 1
27             else:
28                 break
29     if r == l:
30         print(-1)
31     else:
32         print('%d %d' % (l, r))
33

```

代码运行截图

General									
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
229526339	Practice: MinghengZhong	1793C - 44	Python 3	Accepted	171 ms	26076 KB	2023-10-24 11:00:59	2023-10-24 11:00:59	 Compare

→ Source	Copy
<pre> t = int(input()) for _ in range(t): n = int(input()) big = list(map(int, input().split())) num = [0]*big big.sort() l = 1 r = n s = 0 b = n-1 while True: if r == l: break else: if num[r] == big[s]: s += 1 r -= 1 elif num[r] == big[b]: b -= 1 r -= 1 elif num[l] == big[s]: s += 1 l += 1 elif num[l] == big[b]: b -= 1 l += 1 else: break if r == l: print(-1) else: print('%d %d' % (l, r)) </pre>	

2. 选做题目

368B. Sereja and Suffixes

data structures/dp, 1100

<https://codeforces.com/problemset/problem/368/B>

思路：

先把输入的数据倒过来处理，每次处理，在答案list后面添加一个元，如果是新数字，该元就是前一个元加一，否则为前一个元

之后每次在答案list中调用结果输出即可

代码

```
1 n, m = map(int, input().split())
2 a = list(map(int, input().split()))
3 l = [1]
4 used = [False]*(max(a)+1)
5 used[a[-1]] = True
6 for i in range(n-2, -1, -1):
7     if not used[a[i]]:
8         used[a[i]] = True
9         l.append(l[-1]+1)
10    else:
11        l.append(l[-1])
12 for _ in range(m):
13     print(l[-int(input())])
14
```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
229529171	Practice: MinghengZhong	368B - 18	Python 3	Accepted	655 ms	13320 KB	2023-10-24 11:27:12	2023-10-24 11:27:12	★	Compare

→ Source

Copy

```
n, m = map(int, input().split())
a = list(map(int, input().split()))
l = [1]
used = [False]*(max(a)+1)
used[a[-1]] = True
for i in range(n-2, -1, -1):
    if not used[a[i]]:
        used[a[i]] = True
        l.append(l[-1]+1)
    else:
        l.append(l[-1])
for _ in range(m):
    print(l[-int(input())])
```

1764C. Doremy's City Construction

graphs, greedy, 1400

<https://codeforces.com/problemset/problem/1764/C>

思路：

题目的意思是，每个节点必须要比与之连接的所有节点都要大或者小

如果把所有的数分成两部分，其中一组的所有数比另一组的所有数都要大，则从这两组数中分别任选一个，之间都可以连线。因此，连线数就是两组数数量之积，答案就是这个积的最大值。要得到最大值，只需要让两组数数量之差最小，则用双指针从排序后的数组中间开始查找，找到第一次数量变化就退出，从该处将数组分开，两边的数量之积就是答案

例外是所有数都相等，要首先排除，这时只要输出 $n//2$ 即可

代码

```
1  t = int(input())
2  for _ in range(t):
3      n = int(input())
4      a = list(map(int, input().split()))
5      if max(a) == min(a):
6          print(n//2)
7      else:
8          a.sort()
9          if n % 2 == 0:
10             i = n//2-1
11             j = n//2
12         else:
13             i = j = n//2
14         while True:
15             if a[i] != a[i+1]:
16                 A = i+1
17                 break
18             elif a[j] != a[j-1]:
19                 A = j
20                 break
21             else:
22                 i += 1
23                 j -= 1
24         print(A*(n-A))
25
```

代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
229538712	Practice: MinghengZhong	1764C - 29	Python 3	Accepted	171 ms	26456 KB	2023-10-24 12:45:07	2023-10-24 12:45:08	★	<button>Compare</button>

→ Source

Copy

```
t = int(input())
for _ in range(t):
    n = int(input())
    a = list(map(int, input().split()))
    if max(a) == min(a):
        print(n//2)
    else:
        a.sort()
        if n % 2 == 0:
            i = n//2-1
            j = n//2
        else:
            i = j = n//2
        while True:
            if a[i] != a[i+1]:
                A = i+1
                break
            elif a[j] != a[j-1]:
                A = j
                break
        else:
            i += 1
            j -= 1
        print(A*(n-A))
```

3. 学习总结和收获

贪心算法写起来很费脑子，因为要想出一种最贪的策略，并且用代码表达出来。但是一旦想到策略，写起来就很简单。

理论上所有题目都可以用dp枚举出来，但是很容易TLE，很多时候可以把贪心视为一种优化写到dp里面，甚至代替dp。相比一般的剪枝，贪心可以带来很大的优化，但是贪心很难写。

可以概括为：好写的方法慢，快的方法难写。