Optional Assignment: 课程总结

Updated 1640 GMT+8 Dec 28, 2023

2023 fall, Complied by Xinjie Song, Phy

说明：

1）12月28日期末机考： AC6。

2）本次作业可选，不计分。

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。

1. 期末机考题目

27273: 简单的数学题

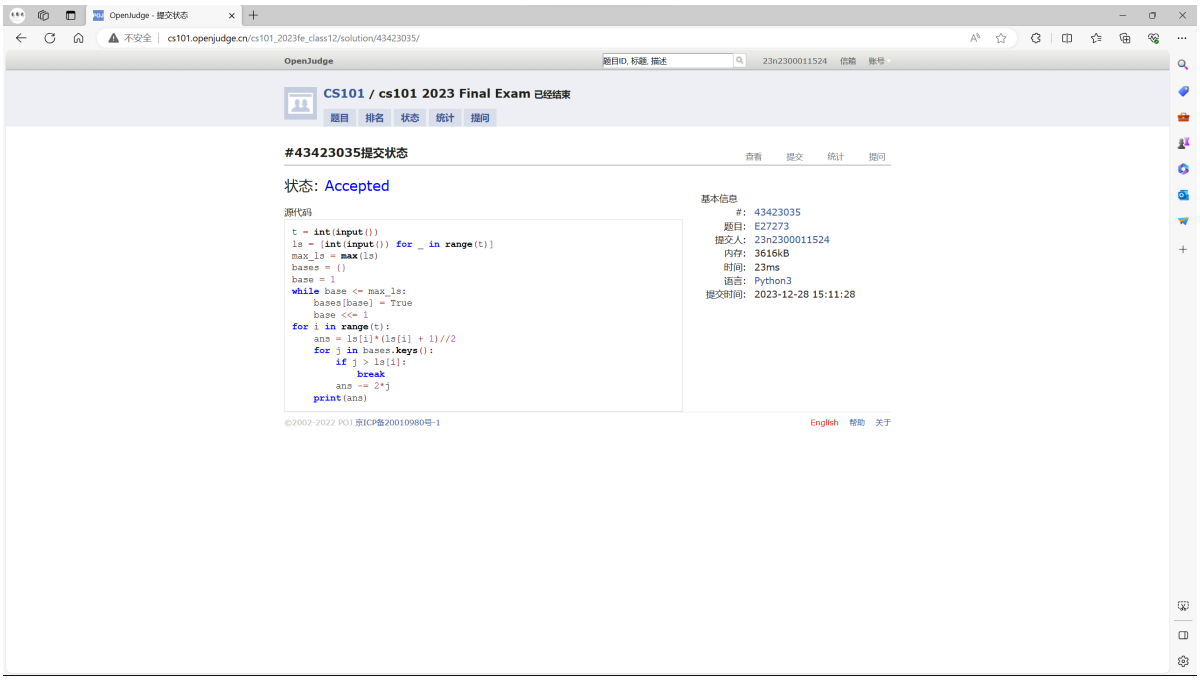
implementation, math, http://cs101.openjudge.cn/practice/27273

思路：本来都写完累加求和了，快提交了看见提示，该成了等差数列求和，再疯狂一点直接把等比数列求和一块写了

代码

```
t = int(input())
ls = [int(input()) for _ in range(t)]
max_ls = max(ls)
bases = {}
base = 1
while base <= max_ls:
    bases[base] = True
    base <<= 1
for i in range(t):
    ans = ls[i]*(ls[i] + 1)//2
    for j in bases.keys():
        if j > ls[i]:
            break
        ans -= 2*j
    print(ans)
```

代码运行截图



27301: 给植物浇水

Implementation, two pointers, http://cs101.openjudge.cn/practice/27301

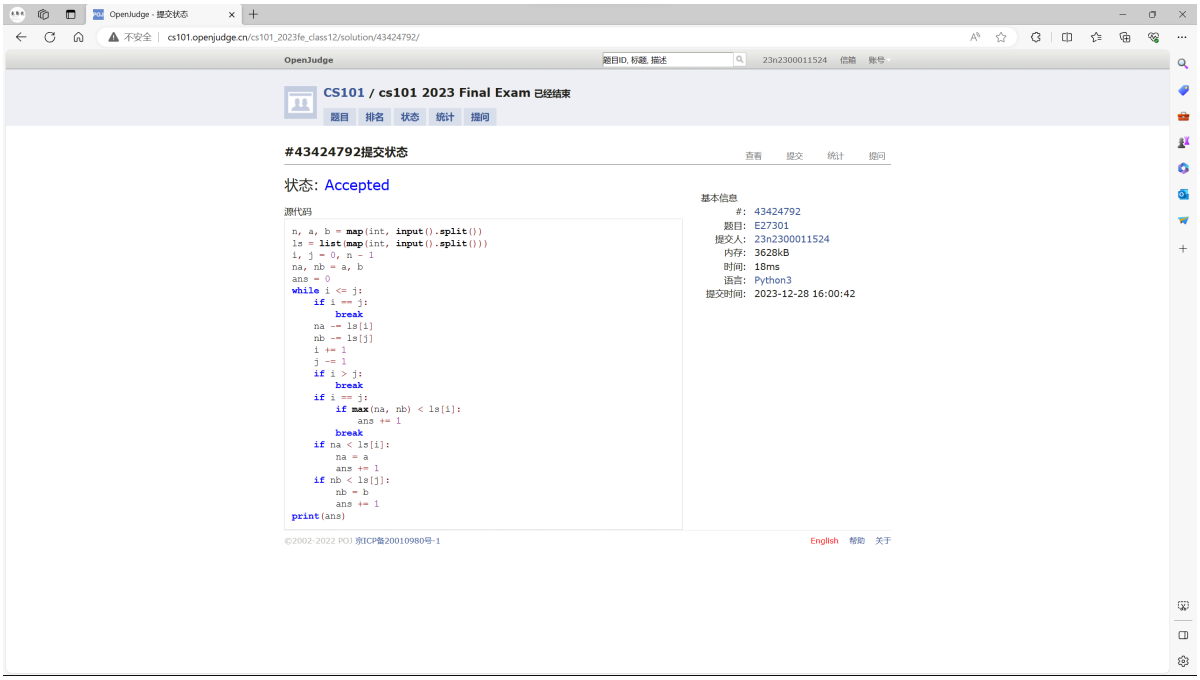
思路：模拟，题目有点歧义，按节约用水的角度理解是正确的

代码

```
n, a, b = map(int, input().split())
ls = list(map(int, input().split()))
i, j = 0, n - 1
na, nb = a, b
ans = 0
while i <= j:
    if i == j:
        break
    na -= ls[i]
    nb -= ls[j]
    i += 1
    j -= 1
    if i > j:
        break
    if i == j:
        if max(na, nb) < ls[i]:
            ans += 1
        break
    if na < ls[i]:
        na = a
        ans += 1
    if nb < ls[j]:
        nb = b
        ans += 1
```

```
print(ans)
```

代码运行截图



27274: 字符串提炼

implementation, http://cs101.openjudge.cn/practice/27274

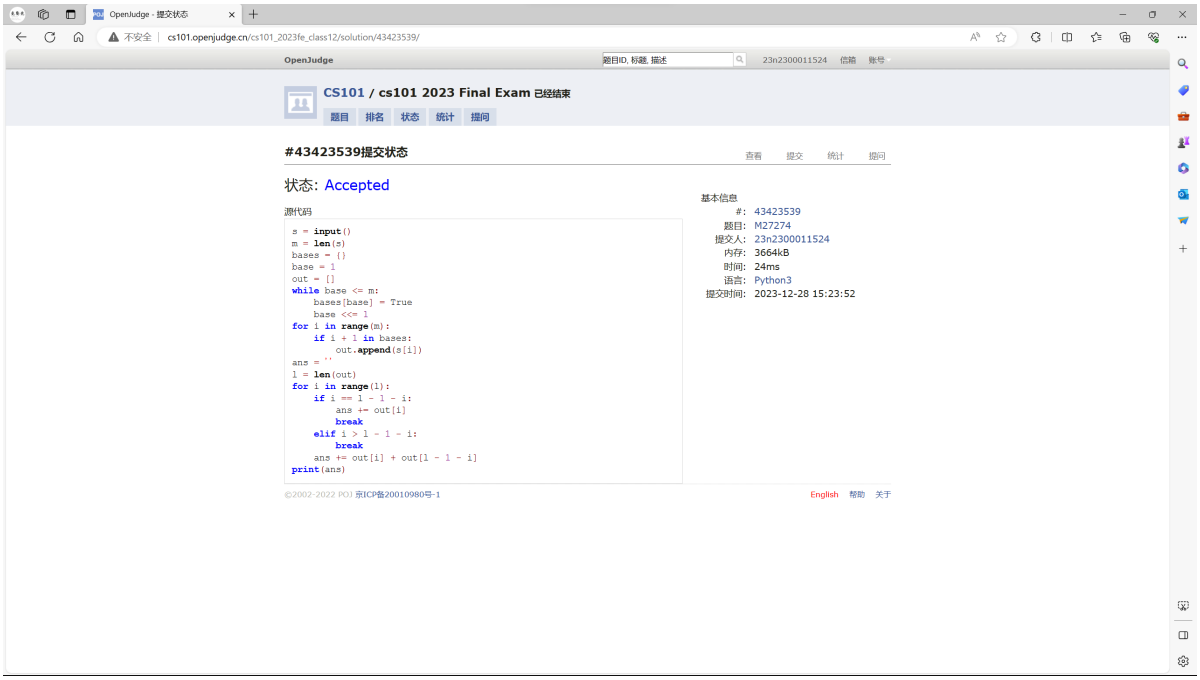
思路：模拟，注意判断最后只剩一个字符的情况

代码

```
s = input()
m = len(s)
bases = {}
base = 1
out = []
while base <= m:
    bases[base] = True
    base <<= 1
for i in range(m):
    if i + 1 in bases:
        out.append(s[i])
ans = ''
l = len(out)
for i in range(l):
    if i == l - 1 - i:
        ans += out[i]
        break
    elif i > l - 1 - i:
        break
```

```
    ans += out[i] + out[l - 1 - i]
print(ans)
```

代码运行截图



27310: 积木

implementation, brute force, http://cs101.openjudge.cn/practice/27310

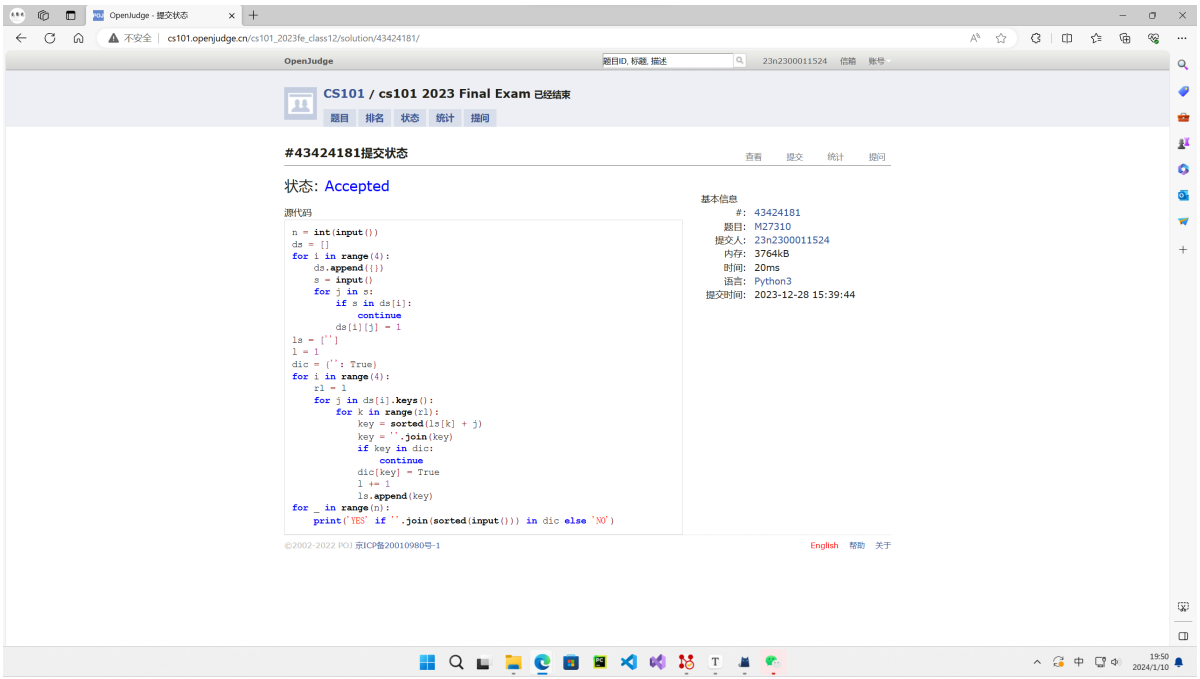
思路：考虑到只有四块积木，枚举所有可能情况用时也是可以接受的，这里要注意如何处理四块积木顺序不同的问题，我采用对序列进行排序的方法，可以确保正确性和简洁性。其中，按积木顺序枚举时，等价于分组背包和桶的结合，当然不用桶也可以，只是会重复处理某些相同序列。

代码

```
n = int(input())
ds = []
for i in range(4):
    ds.append({})
    s = input()
    for j in s:
        if s in ds[i]:
            continue
        ds[i][j] = 1
ls = ['']
l = 1
dic = {'': True}
for i in range(4):
    rl = l
    for j in ds[i].keys():
```

```
        for k in range(rl):
            key = sorted(ls[k] + j)
            key = ''.join(key)
            if key in dic:
                continue
            dic[key] = True
            l += 1
            ls.append(key)
for _ in range(n):
    print('YES' if ''.join(sorted(input())) in dic else 'NO')
```

代码运行截图



27237: 体育游戏跳房子

bfs, http://cs101.openjudge.cn/practice/27237

思路：类似于走山路，这里将步数和扔法同时存入堆，保证最优解。

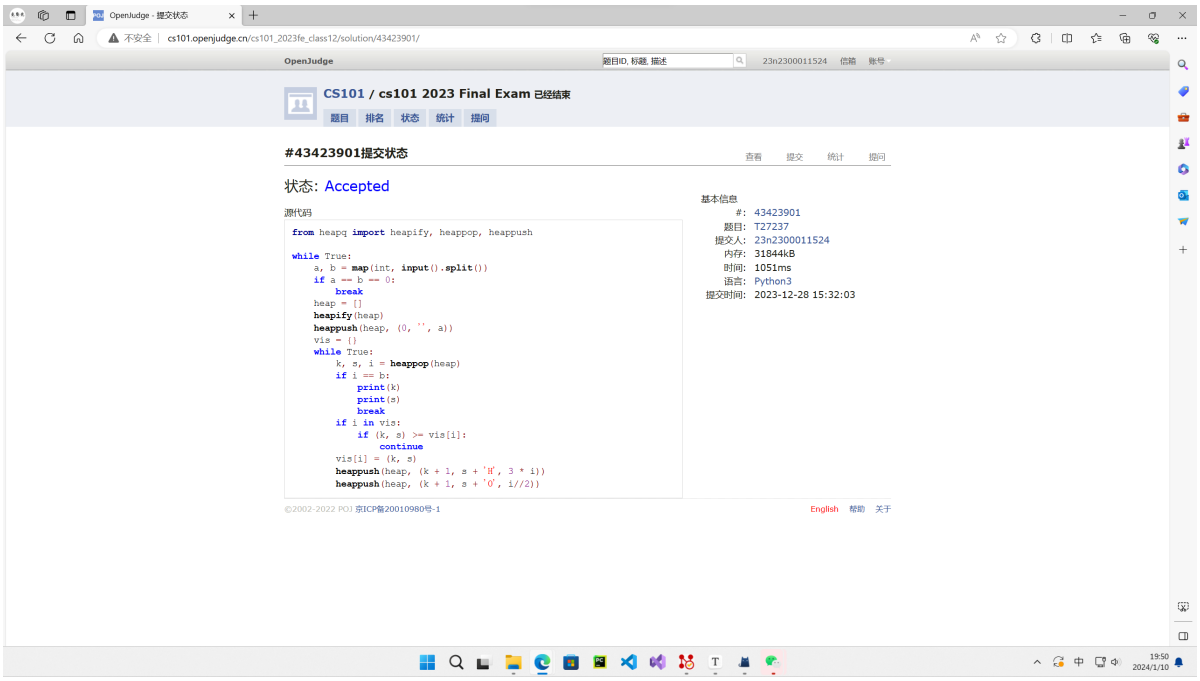
代码

```
from heapq import heapify, heappop, heappush

while True:
    a, b = map(int, input().split())
    if a == b == 0:
        break
    heap = []
    heapify(heap)
    heappush(heap, (0, '', a))
    vis = {}
```

```
    while True:
        k, s, i = heappop(heap)
        if i == b:
            print(k)
            print(s)
            break
        if i in vis:
            if (k, s) >= vis[i]:
                continue
        vis[i] = (k, s)
        heappush(heap, (k + 1, s + 'H', 3 * i))
        heappush(heap, (k + 1, s + 'O', i//2))
```

代码运行截图



27373: 最大整数

dp, greedy, string, sort, http://cs101.openjudge.cn/practice/27373

思路：借鉴最大新整数的思路，自定义比较函数首先对数据进行处理，然后读到要求从中选取几个组成一个位数不超过m的新正整数很自然地联想到背包问题，这里答案就显而易见了

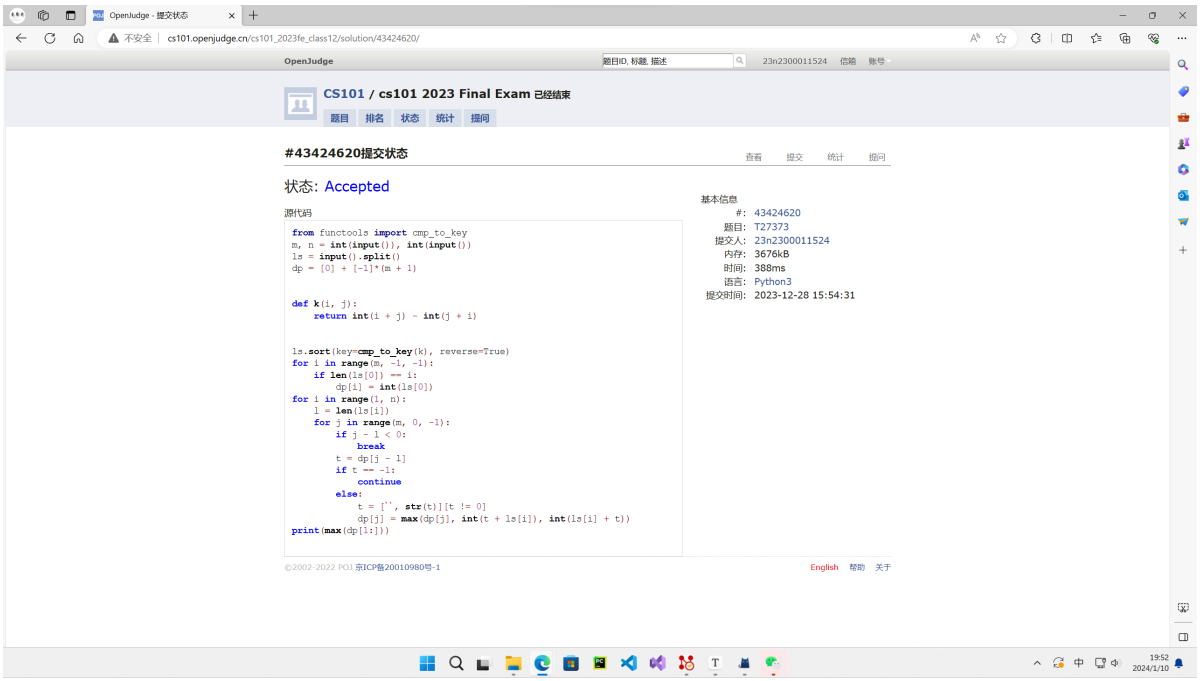
代码

```
from functools import cmp_to_key
m, n = int(input()), int(input())
ls = input().split()
dp = [0] + [-1]*(m + 1)
```

```
def k(i, j):
    return int(i + j) - int(j + i)


ls.sort(key=cmp_to_key(k), reverse=True)
for i in range(m, -1, -1):
    if len(ls[0]) == i:
        dp[i] = int(ls[0])
for i in range(1, n):
    l = len(ls[i])
    for j in range(m, 0, -1):
        if j - l < 0:
            break
        t = dp[j - l]
        if t == -1:
            continue
        else:
            t = ['', str(t)][t != 0]
            dp[j] = max(dp[j], int(t + ls[i]), int(ls[i] + t))
print(max(dp[1:]))
```

代码运行截图



2. 课程总结

10号终于考完了所有期末考试，来写一写OptionalAssignment。

本人水平难以和众佬相比，感觉没被选上的Trouble题目如果选了或者贪心题目那我就告别AC6了；众佬们讨论的难题大部分我是一点思路没有，临近期末斗胆挑战的CF1875D是思修课两小时想出来的，其中一小时想出来解法，一小时完成剪枝，如果放在考试简直不敢想象。至于线段树和树状数组至多会用，难以理解。

至于"基础"这个东西，说有，我在初中学过C和C++，高考后的暑假学了Python；说没有，我只学了语法，算法只会递归，贪心、动态规划、图搜索什么的都是选课以后才知道要去学的，学习方式就是看书和写题，当然最重要的还是要理解每种算法的核心，比如理解动态规划算法，不是背下来背包问题的模板，而是要理解动态规划算法通过避免重复求解相同子问题减小时间复杂度，这样才能在面对不是背包问题的题目时游刃有余。学算法要小心"Accepted"的甜蜜的陷阱，切忌沉迷于找语法题或简单的算法题一边过的快乐中（那种不WA十次过不了的变态语法题除外，不过这种题的用处一言难尽……）。此外，我非常赞同"很久想不出来就看答案"这个行为，因为不看提示或者答案干想收益很小。

《算法图解》是非常好的一本书，插图非常有助于理解，在没听课的前提下（划掉）。闫老师的自编教材随着讲解配套有题目，很适合边学边练。如果想学语法，2023及以前的同学可能需要看教材，但2023年以后的同学可能只看2023fall群里好心人总结的cheat sheet就可以速通语法了。快考试时发现群里有同学分享了这个OI Wiki - OI Wiki (oi-wiki.org)网站，看完背包问题大合集发现这个网站有很多很多算法的讲解，相见恨晚。

计算概论这门课，教会了我善用GPT。高三第一次见到GPT，我以为只是个大号聊天工具和作文作弊工具，后来慢慢听说大学生可以用GPT写作业（小声），再到闫老师推荐用GPT写程序，该程序，我才慢慢体会到GPT的强大之处，我相信用好GPT和编程在今后的学习中肯定帮助非常大，当你要进行繁杂的重复性工作，你会想到用程序来解决问题，你也会进一步想到不会的语法（如表格处理、文件处理等）GPT总是会的，那你就把自己解放出来了——这是最初步的应用。进一步地，以物理为例，编程可以用来计算，模拟，等等。学习计算概论这门课，对于非信科的同学，我认为，重要的不是学会编程，期末AC几道题，而是要学会利用计算机这个工具。提到AI和课程群，我认为有必要开发一个AI小助手负责总结每天同学们讨论了哪些算法、哪些题目，不然看着动辄999+的消息实在令人害怕……

最后，我真的只是打字快一点。考前还在担心机房的键盘不好用会不会影响发挥，看到可以自带键盘便抱上我的珂芝k75金粉轴直奔考场；戏剧性的一幕是，居然真的有人带了把青轴去考试，考前群里的预言成真了。不知不觉间，作业编号就从0排到了F（刚开始是A我以为只是其中过后换了个编号方式，后来意识到可能是16进制编号），看着考完机考渐渐冷清课程群有种说不出的感觉，每天打开网站看看有没有会做的每日选做题目的日子过去了，后面有时间可能还是会写两道的，但会有时间吗？最后的最后，非常幸运在还没扩名额的时候选上实验班，考笔试真的受不了，怎么能把人脑当CPU用呢；非常感谢闫老师和助教的付出，课程群实在太热闹了。

跟风附上自编cheat sheet和写过的题的代码（可能没什么用，因为我没写注释；但好在变量名字不是某神秘主义，辅以GPT应该是能看懂的；当然，最好理解的还是中文变量名，非常神奇）。