

Assignment #D: Mock Exam下元节

Updated 1729 GMT+8 Dec 4, 2025

2025 fall, Compiled by 郑文萱 基础医学院

说明:

1. Dec月考: AC3 (请改为同学的通过数)。考试题目都在“题库(包括计概、数算题目)”里面,按照数字题号能找到,可以重新提交。作业中提交自己最满意版本的代码和截图。
2. 解题与记录: 对于每一个题目, 请提供其解题思路(可选), 并附上使用Python或C++编写的源代码(确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted)。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。(推荐使用Typora <https://typoraio.cn> 进行编辑, 当然你也可以选择Word。)无论题目是否已通过, 请标明每个题目大致花费的时间。
3. 提交安排: 提交时, 请首先上传PDF格式的文件, 并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像, 提交的文件为PDF格式, 并且“作业评论”区包含上传的.md或.doc附件。
4. 延迟提交: 如果你预计无法在截止日期前提交作业, 请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业, 以保证顺利完成课程要求。

1. 题目

E29945:神秘数字的宇宙旅行

implementation, <http://cs101.openjudge.cn/practice/29945>

思路: 这个是角谷猜想, 没有任何区别。签到题。

代码

```
n=int(input())
while n>1:
    if n%2==1:
        print(f'{n}*3+1={n*3+1}')
        n=n*3+1
    else:
        print(f'{n}/2={n//2}')
        n//=2
print('End')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=int(input())
while n>1:
    if n%2==1:
        print(f' {n}*3+1={n*3+1} ' )
        n=n*3+1
    else:
        print(f' {n}/2={n//2} ' )
        n//=2
print(' End ')
```

©2002-2022 POJ 京ICP备20010980号-1

E29946:删数问题

monotonic stack, greedy, <http://cs101.openjudge.cn/practice/29946>

思路: 昨天刚做了一个单调栈, 但其实不太掌握啊, 所以就用普通方法做了。

代码

```
n=list(map(int,input()))
k=int(input())
idx=0
output=[]
while len(n)-k>0:
    minwei = float('inf')
    for item in n[idx:k+1]:
        if item<minwei:
            idx=n.index(item,idx)
            minwei=item
    k+=1
    idx+=1
    output.append(str(minwei))
print(int(''.join(output)))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=list(map(int,input()))
k=int(input())
idx=0
output=[]
while len(n)-k>0:
    minwei = float('inf')
    for item in n[idx:k+1]:
        if item<minwei:
            idx=n.index(item,idx)
            minwei=item
    k+=1
    idx+=1
    output.append(str(minwei))
print(int(''.join(output)))
```

©2002-2022 POJ 京ICP备20010980号-1

E30091:缺德的图书馆管理员

greedy, <http://cs101.openjudge.cn/practice/30091>

思路: 这个题跟每日选做里面的ants内核一模一样, 看来期末很有可能考每日选做变形题(? , 但我还没做完, 正在做了www。

代码

```
l=int(input())+1
n=int(input())
students=list(map(int,input().split()))
maxt=0
mint=0
for s in students:
    if s>=l/2:
        maxt=max(s,maxt)
        mint=max(l-s,mint)
    else:
        maxt=max(l-s,maxt)
        mint=max(s,mint)
print(f'{mint} {maxt}')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
l=int(input())+1
n=int(input())
students=list(map(int,input().split()))
maxt=0
mint=0
for s in students:
    if s>=l/2:
        maxt=max(s,maxt)
        mint=max(l-s,mint)
    else:
        maxt=max(l-s,maxt)
        mint=max(s,mint)
print(f' {mint} {maxt}')
```

©2002-2022 POJ 京ICP备20010980号-1

M27371:Playfair密码

simulation, string, matrix, <http://cs101.openjudge.cn/practice/27371>

思路: 这个题太长了, 我最后一个小时一直在写这个但拼尽全力无法战胜。考完请教了D老师, 发现用列表推导式还有字典可以很好地处理复杂的代码, 这真是个很考验语法的题。还有直接用集合会返回随机值, 这是可以用OrderedDict保持原先顺序。还可以直接replace(j,i)这样的, 我真的是语法很烂啊(算法也烂)。还有老师上课讲过的ASCII打表方法, 要素过多。

代码

```
from collections import OrderedDict
key = input().replace('j', 'i')
keys = list(OrderedDict.fromkeys(key))

#用chr(ord('a') + i) for i in range(26)打字母表并跳过j
alphabet = [chr(ord('a') + i) for i in range(26) if chr(ord('a') + i) != 'j']
for letter in alphabet:
    if letter not in keys:
        keys.append(letter)

matrix = []
for i in range(5):
    matrix.append(keys[i * 5:(i + 1) * 5])

pos = {} #用字典找位置, 用空间换时间
for i in range(5):
    for j in range(5):
        pos[matrix[i][j]] = (i, j)
```

```
def encrypt(plaintext):
    text = []
    for ch in plaintext:
        if ch == 'j':
            text.append('i')
        else:
            text.append(ch)

    pairs = []
    i = 0
    while i < len(text):
        if i == len(text) - 1:
            pairs.append([text[i], 'x' if text[i] != 'x' else 'q'])
            i += 1
        elif text[i] == text[i + 1]:
            pairs.append([text[i], 'x' if text[i] != 'x' else 'q'])
            i += 1
        else:
            pairs.append([text[i], text[i + 1]])
            i += 2

    cipher = []
    for a, b in pairs:
        row1, col1 = pos[a]
        row2, col2 = pos[b]
        if row1 == row2:
            cipher.append(matrix[row1][(col1 + 1) % 5])
            cipher.append(matrix[row2][(col2 + 1) % 5])
        elif col1 == col2:
            cipher.append(matrix[(row1 + 1) % 5][col1])
            cipher.append(matrix[(row2 + 1) % 5][col2])
        else:
            cipher.append(matrix[row1][col2])
            cipher.append(matrix[row2][col1])

    return ''.join(cipher)

n = int(input())
for _ in range(n):
    plaintext = input()
    print(encrypt(plaintext))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import OrderedDict
key = input().replace('j', 'i')
keys = list(OrderedDict.fromkeys(key))

#用chr(ord('a') + i) for i in range(26) 打字母表并跳过j
alphabet = [chr(ord('a') + i) for i in range(26) if chr(ord('a') + i) != 'j']
for letter in alphabet:
    if letter not in keys:
        keys.append(letter)

matrix = []
for i in range(5):
    matrix.append(keys[i * 5:(i + 1) * 5])

pos = {} #用字典找位置, 用空间换时间
for i in range(5):
    for j in range(5):
        pos[matrix[i][j]] = (i, j)

def encrypt(plaintext):
    text = []
    for ch in plaintext:
        if ch == 'j':
```

T30201:旅行售货商问题

dp,dfs, <http://cs101.openjudge.cn/practice/30201>

思路: 这个题我考试的时候记得在算法图解上提到了, 但没有做, 课下看到同学们说可以固定起点, 价格相同。位运算也挺神奇的, 跟一个个小开关一样, 只有01两种状态, 但目前我做过的相关题目太少, 所以并不是很掌握写法, 期末上机是不是要考啊(,,,)。第一次花了好几天一点一点理解啃下来感觉还是挺难的。

代码

```
n=int(input())
cost=[list(map(int,input().split())) for _ in range(n)]
INF=10**9
dp=[[INF]*n for _ in range(1<<n)] #1向左移动1位就是乘2, 移动n位就是2^n
dp[1][0]=0 #从城市0出发
for mask in range(1 << n):
    for i in range(n):
        if dp[mask][i] == INF:
            continue
        if not (mask >> i & 1): #i必须已访问, 不然mask不成立
```

```

        continue
    for j in range(n):        #尝试访问j
        if mask >> j & 1:    #所以j不能已访问
            continue
        new_mask = mask | (1 << j) #将j设为已访问
        new_cost = dp[mask][i] + cost[i][j]
        if new_cost < dp[new_mask][j]: #用max()好像比较慢, 改成if判断
            dp[new_mask][j] = new_cost

ans = INF
full_mask = (1 << n) - 1
for i in range(n):
    if dp[full_mask][i] != INF:
        total = dp[full_mask][i] + cost[i][0]    #回到起点0
        if total < ans:
            ans = total
print(ans)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n=int(input())
cost=[list(map(int,input().split())) for _ in range(n)]
INF=10**9
dp=[[INF]*n for _ in range(1<<n)] #1向左移动1位就是乘2, 移动n位就是2^n
dp[1][0]=0 #从城市0出发
for mask in range(1 << n):
    for i in range(n):
        if dp[mask][i] == INF:
            continue
        if not (mask >> i & 1): #i必须已访问, 不然mask不成立
            continue
        for j in range(n):        #尝试访问j
            if mask >> j & 1:    #所以j不能已访问
                continue
            new_mask = mask | (1 << j) #将j设为已访问
            new_cost = dp[mask][i] + cost[i][j]
            if new_cost < dp[new_mask][j]: #用max()好像比较慢, 改成if判断
                dp[new_mask][j] = new_cost

ans = INF
full_mask = (1 << n) - 1
for i in range(n):
    if dp[full_mask][i] != INF:
        total = dp[full_mask][i] + cost[i][0]    #回到起点0
        if total < ans:
            ans = total
print(ans)

```

T30204:小P的LLM推理加速

greedy, <http://cs101.openjudge.cn/practice/30204>

思路：考试的时候没看。题意还是比较好理解的，就是两个周期循环，但有点不太清楚的是，题目里面说执行连续的推理任务，我觉得是必须连在一起，但糖果店里又感觉不一定是连续的。由于能耗大小的不确定性，使得排序、判断的程序变多了。最后的思路是：选一个 $x+y$ 最小且 x 最大的主力，其他的按 x 升序，遍历其他小核的 x ，不断与全选主力比较大小，当小核越过主力时退出循环。

代码

```
n,m=map(int,input().split())
battery=[tuple(map(int,input().split())) for _ in range(n)]
battery.sort(key=lambda x:(x[0]+x[1],-x[0]))
main=battery[0][0]+battery[0][1]
other=sorted(battery[1:])
rnd=m//main*2 + int(m%main>=battery[0][0])
s=0
for i in range(n - 1):
    s += other[i][0]
    if m < s or other[i][0]>=main:
        break
    rnd=max(rnd,i+1+(m-s)//main*2+int((m-s)%main>=battery[0][0]))
print(rnd)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n,m=map(int,input().split())
battery=[tuple(map(int,input().split())) for _ in range(n)]
battery.sort(key=lambda x:(x[0]+x[1],-x[0]))
main=battery[0][0]+battery[0][1]
other=sorted(battery[1:])
rnd=m//main*2 + int(m%main>=battery[0][0])
s=0
for i in range(n - 1):
    s += other[i][0]
    if m < s or other[i][0]>=main:
        break
    rnd=max(rnd,i+1+(m-s)//main*2+int((m-s)%main>=battery[0][0]))
print(rnd)
```

©2002-2022 POJ 京ICP备20010980号-1

2. 学习总结和收获

月考AC3，第四题是在能力范围内的；第五题状压dp的了解仅限于在群里看到别的同学问到于是问了一下D老师大概是个啥，但并不会做，位运算也是了解但不掌握；第六题考试的时候没看，但其实不算特别难(?)，对我来说比没学过的状压dp要好理解一些。这次月考的时候有在自己调试代码，在写到中间时就排除了一些小错误。感觉个人能力有限，期末复习准备主攻M题了，而且最近发现又有很多没见过的东西从巨佬同学们嘴里蹦出来，倒地。