

Assignment #C: 图 (2/4)

Updated 2329 GMT+8 Nov 24, 2025

2025 fall, Complied by 同学的姓名、院系

说明:

1. 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。

2. 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。
3. 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

M909. 蛇梯棋

bfs, <https://leetcode.cn/problems/snakes-and-ladders/>

思路: dp

代码:

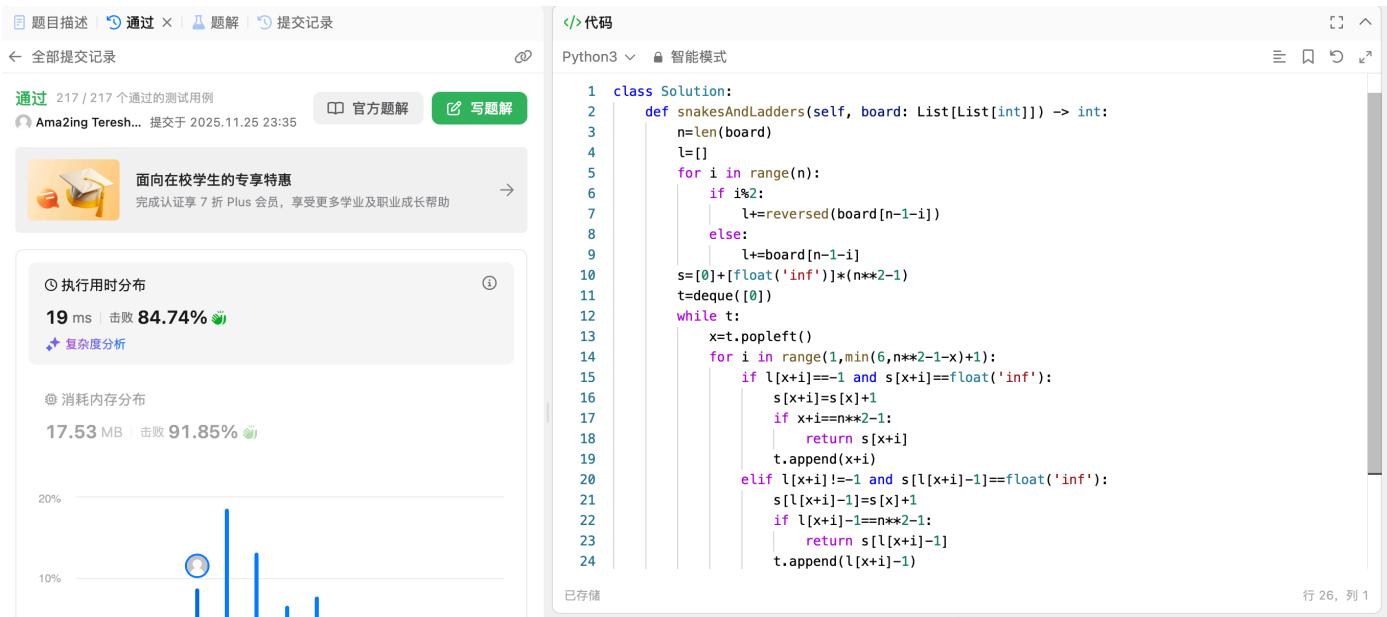
```
class Solution:
    def snakesAndLadders(self, board: List[List[int]]) -> int:
        n=len(board)
        l=[]
        for i in range(n):
            if i%2:
                l+=reversed(board[n-1-i])
            else:
                l+=board[n-1-i]
        s=[0]+[float('inf')]*(n**2-1)
        t=deque([0])
        while t:
```

```

x=t.popleft()
for i in range(1,min(6,n**2-1-x)+1):
    if l[x+i]==-1 and s[x+i]==float('inf'):
        s[x+i]=s[x]+1
    if x+i==n**2-1:
        return s[x+i]
    t.append(x+i)
elif l[x+i]!=-1 and s[l[x+i]-1]==float('inf'):
    s[l[x+i]-1]=s[x]+1
    if l[x+i]-1==n**2-1:
        return s[l[x+i]-1]
    t.append(l[x+i]-1)
return -1

```

代码运行截图 (至少包含有"Accepted")



sy382: 有向图判环 中等

dfs, topological sort, <https://sunnywhy.com/sfbj/10/3/382>

思路：不知道怎么说，反正就是从上往下捋，一张图捋不完就有环

代码：

```

n,m=map(int,input().split())
a=[0 for _ in range(n)]
b={i:[] for i in range(n)}
for _ in range(m):
    x,y=map(int,input().split())
    a[y]+=1

```

```
b[x].append(y)
s=[]
for i in range(n):
    if a[i]==0:
        s.append(i)
def solve(x):
    for y in b[x]:
        a[y]-=1
        if not a[y]:
            solve(y)
for i in s:
    solve(i)
if a==[0]*n:
    print('No')
else:
    print('Yes')
```

代码运行截图 (至少包含有"Accepted")

```
1  n,m=map(int,input().split())
2  a=[0 for _ in range(n)]
3  b={i:[] for i in range(n)}
4  for _ in range(m):
5      x,y=map(int,input().split())
6      a[y]+=1
7      b[x].append(y)
8  s=[]
9  for i in range(n):
10     if a[i]==0:
11         s.append(i)
12 def solve(x):
13     for y in b[x]:
14         a[y]-=1
15         if not a[y]:
16             solve(y)
17 for i in s:
18     solve(i)
19 if a==[0]*n:
20     print('No')
21 -1 -1 -1
```

测试输入

提交结果

历史提交

完美通过

[查看题解](#)

100% 数据通过测试 [详情](#)

运行时长: 0 ms

M28046: 词梯

bfs, <http://cs101.openjudge.cn/practice/28046/>

思路：最开始自己的代码有点慢，用了通配符建图就好多了

代码：

```
def solve():
    from collections import deque, defaultdict
    n=int(input())
    a=defaultdict(list)
    b={}
    for _ in range(n):
        x=input()
        b[x]=[]
        for i in range(4):
            a[x[:i]+'*'+x[i+1:]].append(x)
    for l in a.values():
        for i in range(len(l)):
            for j in range(i+1,len(l)):
                b[l[i]].append(l[j])
                b[l[j]].append(l[i])
    l=set([])
    c,d=input().split()
    s=deque([[c]])
    while s:
        x=s.popleft()
        y=x[-1]
        for z in b[y]:
            if z not in l:
                if z==d:
                    return x+[z]
                s.append(x+[z])
                l.add(z)
    return ['NO']
print(' '.join(solve()))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

def solve():
    from collections import deque, defaultdict
    n=int(input())
    a=defaultdict(list)
    b={}
    for _ in range(n):
        x=input()
        b[x]=[]
        for i in range(4):
            a[x[:i]+'*'+x[i+1:]].append(x)
    for l in a.values():
        for i in range(len(l)):
            for j in range(i+1, len(l)):
                b[l[i]].append(l[j])
                b[l[j]].append(l[i])
    l=set([])
    c,d=input().split()
    s=deque([[c]])
    while s:
        x=s.popleft()
        y=x[-1]
        for z in b[y]:
            if z not in l:
                if z==d:
                    return x+[z]
                s.append(x+[z])
                l.add(z)
    return ['NO']
print(''.join(solve()))

```

基本信息

#: 50998589
 题目: 28046
 提交人: 24n2400011474
 内存: 6592kB
 时间: 49ms
 语言: Python3
 提交时间: 2025-11-25 19:29:20

M433.最小基因变化

bfs, <https://leetcode.cn/problems/minimum-genetic-mutation/>

思路: 上一题我的最开始的思路类似这个, 都是最标准的bfs

代码

```

class Solution:
    def minMutation(self, startGene: str, endGene: str, bank: List[str]) -> int:
        if startGene==endGene:
            return 0
        a=set([])
        b=set(bank)
        s=deque([(startGene, 0)])
        r=('A', 'T', 'G', 'C')
        while s:
            x=s.popleft()
            y=x[0]
            for i in range(8):
                for z in r:
                    if z!=y[i]:

```

```
w=y[:i]+z+y[i+1:]
if w in b and w not in a:
    if w==endGene:
        return x[1]+1
    a.add(w)
    s.append((w,x[1]+1))
return -1
```

代码运行截图 (至少包含有"Accepted")

The figure shows a screenshot of a programming competition interface. At the top, there are tabs for '题目描述' (Problem Description), '通过' (Accepted), '题解' (Solution), and '提交记录' (Submission History). Below these, a '全部提交记录' (All Submission Records) button is visible. The main area is titled '通过' (Accepted) with '20 / 20 个通过的测试用例' (20 / 20 test cases passed). A message from 'Amaing Teresh...' is displayed, dated '提交于 2025.11.25 23:39'. There are buttons for '官方题解' (Official Solution) and '写题解' (Write Solution). A banner for '面向在校学生的专享特惠' (Exclusive benefits for college students) is shown, along with a graduation cap icon. Below this, a section for '① 执行用时分布' (Execution Time Distribution) shows '0 ms | 击败 100.00%' with a green checkmark icon. A '复杂度分析' (Complexity Analysis) button is also present. Another section for '② 消耗内存分布' (Memory Usage Distribution) shows '17.41 MB | 击败 85.18%' with a green checkmark icon. At the bottom, a progress bar indicates '100%'. On the right side, there is a '代码' (Code) tab with 'Python3' selected. The code editor shows the following Python code:

```
1 class Solution:
2     def minMutation(self, startGene: str, endGene: str, bank: List[str]) -> int:
3         if startGene==endGene:
4             return 0
5         a=set([])
6         b=set(bank)
7         s=deque([(startGene,0)])
8         r=(*'ATGC')
9         while s:
10             x=s.popleft()
11             y=x[0]
12             for i in range(8):
13                 for z in r:
14                     if z!=y[i]:
15                         w=y[:i]+z+y[i+1:]
16                         if w in b and w not in a:
17                             if w==endGene:
18                                 a.add(w)
19                                 s.append((w,x[1]+1))
20
21     return -1
```

M05443: 兔子与樱花

Dijkstra, <http://cs101.openjudge.cn/practice/05443/>

思路：标准的Dijkstra

代码

```
import heapq
p=int(input())
d={}
for _ in range(p):
    d[input()]=[]
q=int(input())
for _ in range(q):
    x,y,z=input().split()
    d[x].append((int(z),y))
    d[y].append((int(z),x))
r=int(input())
def solve(x,y):
    if y==x:
        return x
    f={x:x}
    while y in f:
        y=f[y]
    return y
```

```

s=[(0,x,_,_)]
heapq.heapify(s)
while s:
    a,b,c,e=heapq.heappop(s)
    if b not in f:
        f[b]=f[c]+'->('+str(e)+')->'+b
        if b==y:
            return f[b]
    for g,h in d[b]:
        if h not in f:
            s.append((a+g,h,b,g))
for _ in range(r):
    x,y=input().split()
    print(solve(x,y))

```

代码运行截图 (至少包含有"Accepted")

#50999285提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

import heapq
p=int(input())
d={}
for _ in range(p):
    d[input()]=[]
q=int(input())
for _ in range(q):
    x,y,z=input().split()
    d[x].append((int(z),y))
    d[y].append((int(z),x))
r=int(input())
def solve(x,y):
    if y==x:
        return x
    f={x:x}
    s=[(0,x,_,_)]
    heapq.heapify(s)
    while s:
        a,b,c,e=heapq.heappop(s)
        if b not in f:
            f[b]=f[c]+'->('+str(e)+')->'+b
            if b==y:
                return f[b]
        for g,h in d[b]:
            if h not in f:
                s.append((a+g,h,b,g))
for _ in range(r):
    x,y=input().split()
    print(solve(x,y))

```

基本信息

#: 50999285
 题目: 05443
 提交人: 24n2400011474
 内存: 3592kB
 时间: 23ms
 语言: Python3
 提交时间: 2025-11-25 19:55:23

M28050: 骑士周游

dfs, <http://cs101.openjudge.cn/practice/28050/>

思路：需要利用华尔多夫规则

代码：

```
n=int(input())
x,y=map(int,input().split())
d={}
r=(( -2, -1), ( -2, 1), ( -1, -2), ( -1, 2), ( 1, -2), ( 1, 2), ( 2, -1), ( 2, 1))
for i in range(n):
    for j in range(n):
        d[(i,j)]=[]
        for di,dj in r:
            if 0<=i+di< n and 0<=j+dj< n:
                d[(i,j)].append((i+di,j+dj))

def solve(l,p,x,y,s):
    if s==n**2-1:
        return True
    q=sorted(d[(x,y)],key=lambda x: p[x[0]][x[1]])
    for a,b in q:
        if not l[a][b]:
            l[a][b]=1
            for i,j in d[(a,b)]:
                p[i][j]-=1
            if solve(l,p,a,b,s+1):
                return True
            for i,j in d[(a,b)]:
                p[i][j]+=1
            l[a][b]=0
    return False
l=[[0]*n for _ in range(n)]
l[x][y]=1
p=[[len(d[(i,j)]) for j in range(n)] for i in range(n)]
for i,j in d[(x,y)]:
    p[i][j]-=1
if solve(l,p,x,y,0):
    print('success')
else:
    print('fail')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n=int(input())
x,y=map(int,input().split())
d={}
r=(-2,-1),(-2,1),(-1,-2),(-1,2),(1,-2),(1,2),(2,-1),(2,1)
for i in range(n):
    for j in range(n):
        d[(i,j)]=[]
        for di,dj in r:
            if 0<=i+di<=n and 0<=j+dj<=n:
                d[(i,j)].append((i+di,j+dj))
def solve(l,p,x,y,s):
    if s==n**2-1:
        return True
    q=sorted(d[(x,y)],key=lambda x: p[x[0]][x[1]])
    for a,b in q:
        if not l[a][b]:
            l[a][b]=1
            for i,j in d[(a,b)]:
                p[i][j]=1
            if solve(l,p,a,b,s+1):
                return True
            for i,j in d[(a,b)]:
                p[i][j]=0
            l[a][b]=0
    return False
l=[[0]*n for _ in range(n)]
l[x][y]=1
p=[[len(d[(i,j)]) for j in range(n)] for i in range(n)]
for i,j in d[(x,y)]:
    p[i][j]-=1
if solve(l,p,x,y,0):
    print('success')
else:
    print('fail')

```

基本信息

#: 50999923
 题目: 28050
 提交人: 24n2400011474
 内存: 4140kB
 时间: 27ms
 语言: Python3
 提交时间: 2025-11-25 20:18:52

2. 学习总结和个人收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025fall每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

每日选做在跟，最近ai用的比较多（为了节省时间debug），希望下次月考不要发挥失常