assignmentC

Assignment #C: 图 (2/4)

Updated 2329 GMT+8 Nov 24, 2025

2025 fall, Complied by 杨知进 物理学院

> **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：

代码：

```
class Solution:
    def snakesAndLadders(self, board: List[List[int]]) -> int:
        n = len(board)
        vis = [False] * (n * n + 1)
        vis[1] = True
        q = [1]
        step = 0
        while q:
```

```
            tmp = q
            q = []
            for x in tmp:
                if x == n * n:
                    return step
                for y in range(x + 1, min(x + 6, n * n) + 1):
                    r, c = (y - 1)//n , (y-1)%n
                    if r % 2:
                        c = n - 1 - c
                    nxt = board[-1 - r][c]
                    if nxt < 0:
                        nxt = y
                    if not vis[nxt]:
                        vis[nxt] = True
                        q.append(nxt)
            step += 1
        return -1
```

代码运行截图 （至少包含有"Accepted"）

Python3 智能模式

```
        while q:
            tmp = q
            q = []
            for x in tmp:
                if x == n * n:
                    return step
                for y in range(x + 1, min(x + 6, n * n) + 1):
                    r, c = (y - 1)//n , (y-1)%n
                    if r % 2:
                        c = n - 1 - c
                    nxt = board[-1 - r][c]
                    if nxt < 0:
                        nxt = y
                    if not vis[nxt]:
                        vis[nxt] = True
                        q.append(nxt)
            step += 1
        return -1
```

已存储

题目描述 | 题解 | 通过 × | 提交记录

← 全部提交记录

通过 217 / 217 个通过的测试用例

image.png

sy382: 有向图判环 中等

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

思路：

拓扑排序，count=n表明没有环

代码：

```
from collections import deque
n, m = map(int,input().split())
adj = [[] for _ in range(n)]
in_degree = [0] * n
for i in range(m):
```

```
    u, v = map(int,input().split())
    adj[u].append(v)
    in_degree[v] += 1
queue = deque([i for i in range(n) if in_degree[i] == 0])
count = 0    while queue:
    node = queue.popleft()
    count+= 1
for neighbor in adj[node]:
        in_degree[neighbor] -= 1
        if in_degree[neighbor] == 0:
            queue.append(neighbor)
if count == n:
    print('No')
else:
    print('Yes')
```

代码运行截图 （至少包含有"Accepted"）

image-1.png

M28046: 词梯

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

思路：

代码：

```
import sys
from collections import deque
from collections import defaultdict
def build_buckets(word):
    for j in range(len(word)):
        bucket = f'{word[:j]}_{word[j+1:]}'
        buckets.setdefault(bucket, set()).add(word)

data=sys.stdin.read()
data=list(data.split())
index=0
n=int(data[index])
index+=1
buckets=defaultdict(set)
for i in range(n):
    word=data[index]
    index+=1
    build_buckets(word)
del_list=[]
for bucket in buckets:
    if len(buckets[bucket]) == 1:
        del_list.append(bucket)
for bucket in del_list:
    buckets.pop(bucket)
begin,end=data[index],data[index+1]
finish_set=set()
queue=deque([[begin]])
while queue:
    path=queue.popleft()
    word=path[-1]
    for i in range(len(word)):
        bucket=f'{word[:i]}_{word[i+1:]}'
        if bucket not in finish_set:
            for new in buckets[bucket]:
                if new == end:
                    path.append(new)
                    print(*path)
                    exit()
                if new != word:
                    queue.append(path+[new])
```

```
            finish_set.add(bucket)
print('NO')
```

代码运行截图 （至少包含有"Accepted"）

#51108651提交状态

状态：Accepted

image-6.png

M433.最小基因变化

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

思路：

代码

```
class Solution:
    def minMutation(self, start: str, end: str, bank: List[str]) -> int:
        if start == end:
            return 0
        bank = set(bank)
        if end not in bank:
            return -1
        q = deque([(start, 0)])
        while q:
            cur, step = q.popleft()
            for i, x in enumerate(cur):
                for y in "ACGT":
                    if y != x:
                        nxt = cur[:i] + y + cur[i + 1:]
                        if nxt in bank:
                            if nxt == end:
                                return step + 1
                            bank.remove(nxt)
                            q.append((nxt, step + 1))
        return -1
```

代码运行截图（至少包含有"Accepted"）

```python
class Solution:
    def minMutation(self, start: str, end: str, bank: List[str]) -> i
        if start == end:
            return 0
        bank = set(bank)
        if end not in bank:
            return -1
        q = deque([(start, 0)])
        while q:
            cur, step = q.popleft()
            for i, x in enumerate(cur):
                for y in "ACGT":
                    if y != x:
                        nxt = cur[:i] + y + cur[i + 1:]
                        if nxt in bank:
                            if nxt == end:
                                return step + 1
                            bank.remove(nxt)
                            q.append((nxt, step + 1))
```

目描述 | 通过 × | 题解 | 提交记录

邹提交记录

通过 20 / 20 个通过的测试用例

image-2.png

M05443: 兔子与樱花

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

思路：

代码

```
```

代码运行截图（至少包含有"Accepted"）

M28050: 骑士周游

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

思路：

启发式搜索，先尝试下一步有更多可能的路径

代码：

```
def dfs(x,y,n,cnt,visited_matrix):
    if cnt == n*n-1:
        return True
    for num,new_x,new_y in build_target(x,y,n,visited_matrix):
        visited_matrix[new_x][new_y] = True
        if dfs(new_x,new_y,n,cnt + 1,visited_matrix):
            return True
        visited_matrix[new_x][new_y] = False
    return False

def build_target(x,y,n,visited_matrix):
    target_list = []
    for dx,dy in delta:
        if 0<=x+dx<n and 0<=y+dy<n and not visited_matrix[x+dx][y+dy]:
            num=0
            for di,dj in delta:
                if 0<=x+dx+di<n and 0<=y+dy+dj<n and not
visited_matrix[x+dx+di][y+dy+dj]:
                    num+=1
            target_list.append([num,x+dx,y+dy])
    target_list.sort()
    return target_list

n=int(input())
start_x,start_y=map(int,input().split())
visited_matrix=[[False]*n for _ in range(n)]
visited_matrix[start_x][start_y] = True
delta=[(1,2),(2,1),(-1,2),(-2,1),(1,-2),(2,-1),(-1,-2),(-2,-1)]
if dfs(start_x,start_y,n,0,visited_matrix):
    print("success")
else:
    print("fail")
```

代码运行截图 （至少包含有"Accepted"）

#51108660提交状态

状态：Accepted

image-8.png

2. 学习总结和个人收获

正在归类bfs的各种应用，在框架之外惊叹于这样基本的算法能做到这么多事情。