Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 刘家亦，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++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），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。

2）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"区有上传的md或者doc附件。

3）如果不能在截止前提交作业，请写明原因。

1. 题目

18160: 最大连通域面积

dfs similar, http://cs101.openjudge.cn/practice/18160

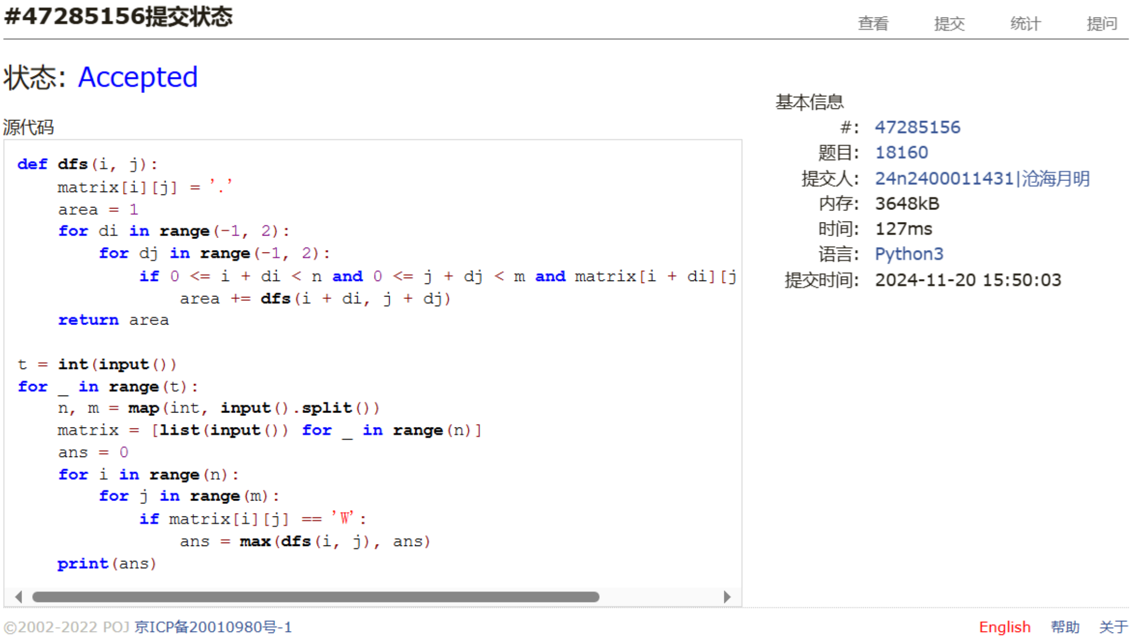
思路：

代码：

```python
def dfs(i, j):
    matrix[i][j] = '.'
    area = 1
    for di in range(-1, 2):
        for dj in range(-1, 2):
            if 0 <= i + di < n and 0 <= j + dj < m and matrix[i + di][j + dj] == 'W':
                area += dfs(i + di, j + dj)
    return area

t = int(input())
for _ in range(t):
    n, m = map(int, input().split())
    matrix = [list(input()) for _ in range(n)]
    ans = 0
    for i in range(n):
        for j in range(m):
            if matrix[i][j] == 'W':
                ans = max(dfs(i, j), ans)
    print(ans)
```

代码运行截图 （至少包含有"Accepted"）



#47285156提交状态

状态: Accepted

源代码

```
def dfs(i, j):
    matrix[i][j] = '.'
    area = 1
    for di in range(-1, 2):
        for dj in range(-1, 2):
            if 0 <= i + di < n and 0 <= j + dj < m and matrix[i + di][j
                area += dfs(i + di, j + dj)
    return area

t = int(input())
for _ in range(t):
    n, m = map(int, input().split())
    matrix = [list(input()) for _ in range(n)]
    ans = 0
    for i in range(n):
        for j in range(m):
            if matrix[i][j] == 'W':
                ans = max(dfs(i, j), ans)
    print(ans)
```

基本信息

#:	47285156
题目:	18160
提交人:	24n2400011431|沧海月明
内存:	3648kB
时间:	127ms
语言:	Python3
提交时间:	2024-11-20 15:50:03

©2002-2022 POJ 京ICP备20010980号-1

19930: 寻宝

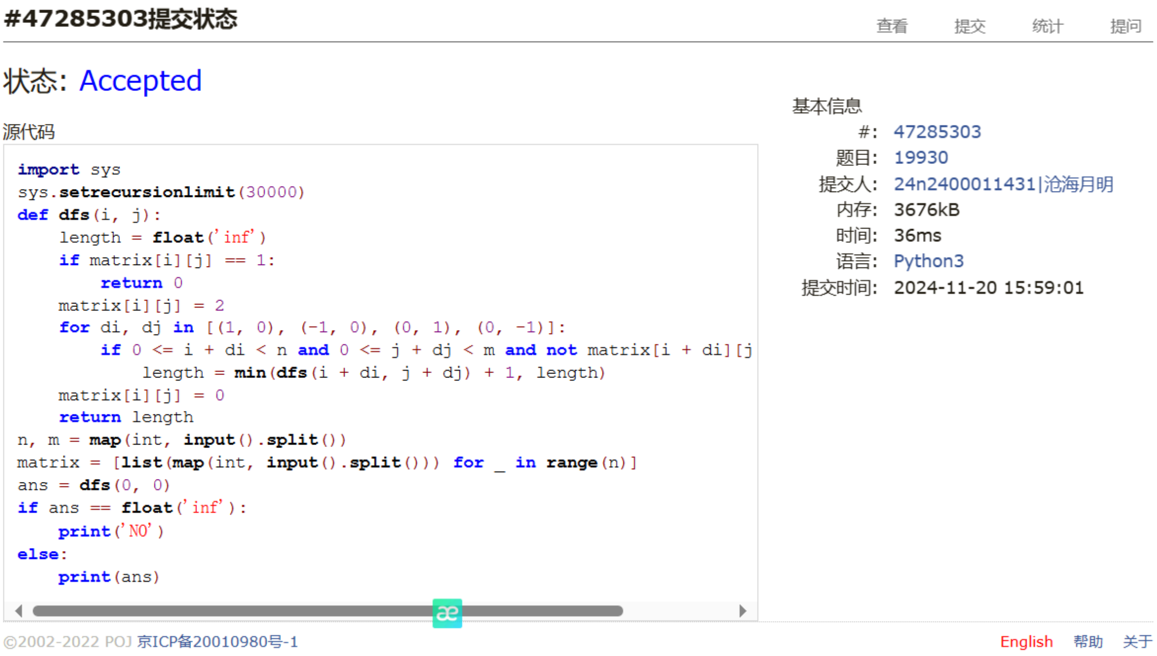
bfs, http://cs101.openjudge.cn/practice/19930

思路:

代码:

```
import sys
sys.setrecursionlimit(30000)
def dfs(i, j):
    length = float('inf')
    if matrix[i][j] == 1:
        return 0
    matrix[i][j] = 2
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if 0 <= i + di < n and 0 <= j + dj < m and not matrix[i + di][j + dj] == 2:
            length = min(dfs(i + di, j + dj) + 1, length)
    matrix[i][j] = 0
    return length
n, m = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(n)]
ans = dfs(0, 0)
if ans == float('inf'):
    print('NO')
else:
    print(ans)
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```
import sys
sys.setrecursionlimit(30000)
def dfs(i, j):
    length = float('inf')
    if matrix[i][j] == 1:
        return 0
    matrix[i][j] = 2
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if 0 <= i + di < n and 0 <= j + dj < m and not matrix[i + di][j
            length = min(dfs(i + di, j + dj) + 1, length)
    matrix[i][j] = 0
    return length
n, m = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(n)]
ans = dfs(0, 0)
if ans == float('inf'):
    print('NO')
else:
    print(ans)
```

基本信息

#:	47285303
题目:	19930
提交人:	24n2400011431|沧海月明
内存:	3676kB
时间:	36ms
语言:	Python3
提交时间:	2024-11-20 15:59:01

04123: 马走日

dfs, http://cs101.openjudge.cn/practice/04123

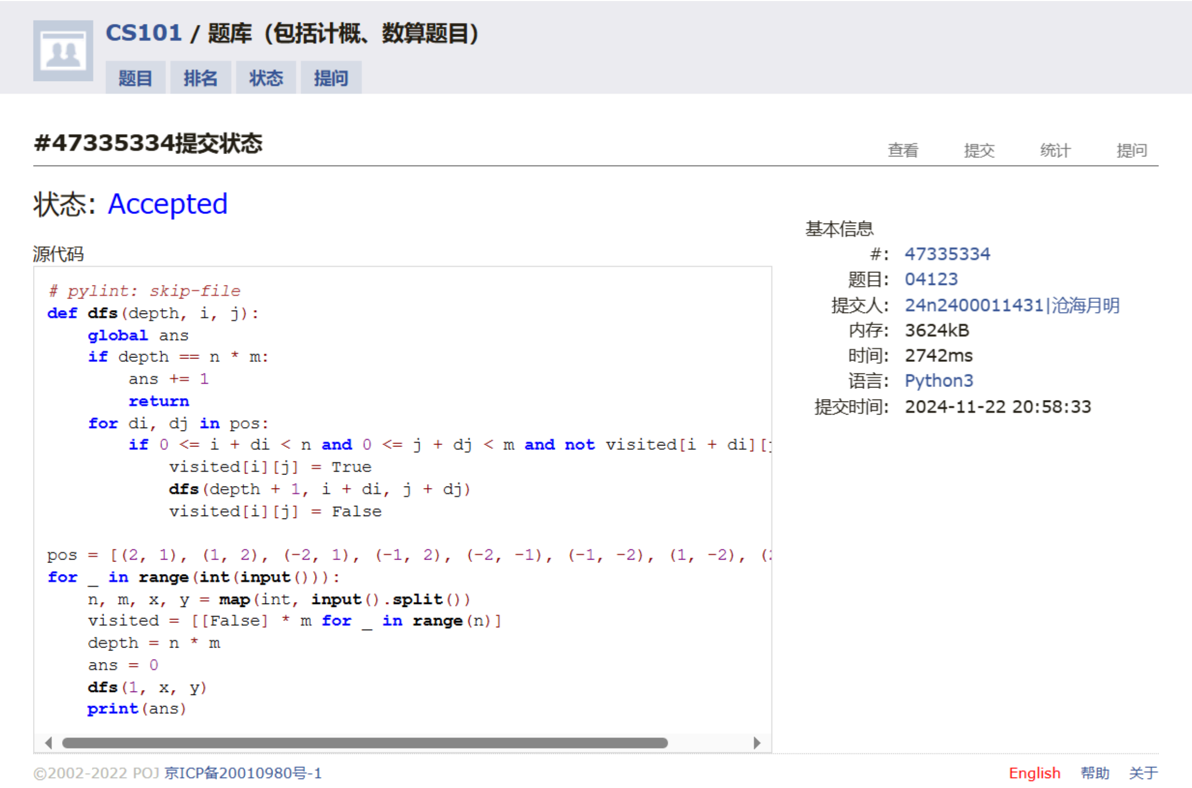
思路：depth可以作为dfs的参数，并且注意要恢复原始状态

代码:

```python
# pylint: skip-file
def dfs(depth, i, j):
    global ans
    if depth == n * m:
        ans += 1
        return
    for di, dj in pos:
        if 0 <= i + di < n and 0 <= j + dj < m and not visited[i + di][j + dj]:
            visited[i][j] = True
            dfs(depth + 1, i + di, j + dj)
            visited[i][j] = False

pos = [(2, 1), (1, 2), (-2, 1), (-1, 2), (-2, -1), (-1, -2), (1, -2), (2, -1)]
for _ in range(int(input())):
    n, m, x, y = map(int, input().split())
    visited = [[False] * m for _ in range(n)]
    depth = n * m
    ans = 0
    dfs(1, x, y)
    print(ans)
```

代码运行截图 （至少包含有"Accepted"）



#47335334提交状态

状态: Accepted

源代码

```
# pylint: skip-file
def dfs(depth, i, j):
    global ans
    if depth == n * m:
        ans += 1
        return
    for di, dj in pos:
        if 0 <= i + di < n and 0 <= j + dj < m and not visited[i + di][
            visited[i][j] = True
            dfs(depth + 1, i + di, j + dj)
            visited[i][j] = False

pos = [(2, 1), (1, 2), (-2, 1), (-1, 2), (-2, -1), (-1, -2), (1, -2), (
for _ in range(int(input())):
    n, m, x, y = map(int, input().split())
    visited = [[False] * m for _ in range(n)]
    depth = n * m
    ans = 0
    dfs(1, x, y)
    print(ans)
```

基本信息

#: 47335334
题目: 04123
提交人: 24n2400011431|沧海月明
内存: 3624kB
时间: 2742ms
语言: Python3
提交时间: 2024-11-22 20:58:33

sy316: 矩阵最大权值路径

dfs, https://sunnywhy.com/sfbj/8/1/316

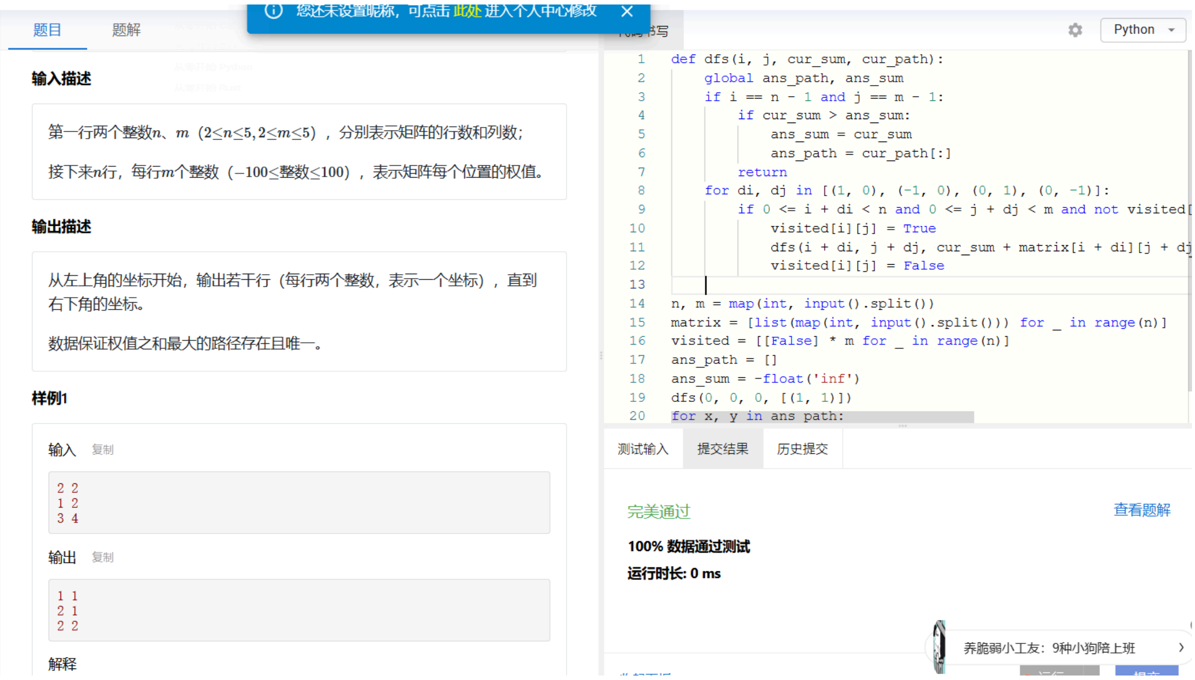
思路:

代码:

```
def dfs(i, j, cur_sum, cur_path):
    global ans_path, ans_sum
    if i == n - 1 and j == m - 1:
        if cur_sum > ans_sum:
            ans_sum = cur_sum
            ans_path = cur_path[:]
        return
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if 0 <= i + di < n and 0 <= j + dj < m and not visited[i + di][j + dj]:
            visited[i][j] = True
            dfs(i + di, j + dj, cur_sum + matrix[i + di][j + dj], cur_path + [(i + di + 1, j + dj + 1)])
            visited[i][j] = False

n, m = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(n)]
visited = [[False] * m for _ in range(n)]
ans_path = []
ans_sum = -float('inf')
dfs(0, 0, 0, [(1, 1)])
for x, y in ans_path:
    print(x, y)
```

代码运行截图 （至少包含有"Accepted"）



您还未设置昵称，可点击 此处 进入个人中心修改

题目 题解

输入描述

第一行两个整数n、m（$2 \le n \le 5, 2 \le m \le 5$），分别表示矩阵的行数和列数；

接下来n行，每行m个整数（$-100 \le 整数 \le 100$），表示矩阵每个位置的权值。

输出描述

从左上角的坐标开始，输出若干行（每行两个整数，表示一个坐标），直到右下角的坐标。

数据保证权值之和最大的路径存在且唯一。

样例1

输入 复制

```
2 2
1 2
3 4
```

输出 复制

```
1 1
2 1
2 2
```

解释

Python

```python
def dfs(i, j, cur_sum, cur_path):
    global ans_path, ans_sum
    if i == n - 1 and j == m - 1:
        if cur_sum > ans_sum:
            ans_sum = cur_sum
            ans_path = cur_path[:]
        return
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if 0 <= i + di < n and 0 <= j + dj < m and not visited[
            visited[i][j] = True
            dfs(i + di, j + dj, cur_sum + matrix[i + di][j + dj
            visited[i][j] = False

n, m = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(n)]
visited = [[False] * m for _ in range(n)]
ans_path = []
ans_sum = -float('inf')
dfs(0, 0, 0, [(1, 1)])
for x, y in ans path:
```

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

运行时长: 0 ms

养脆弱小工友：9种小狗陪上班

LeetCode62.不同路径

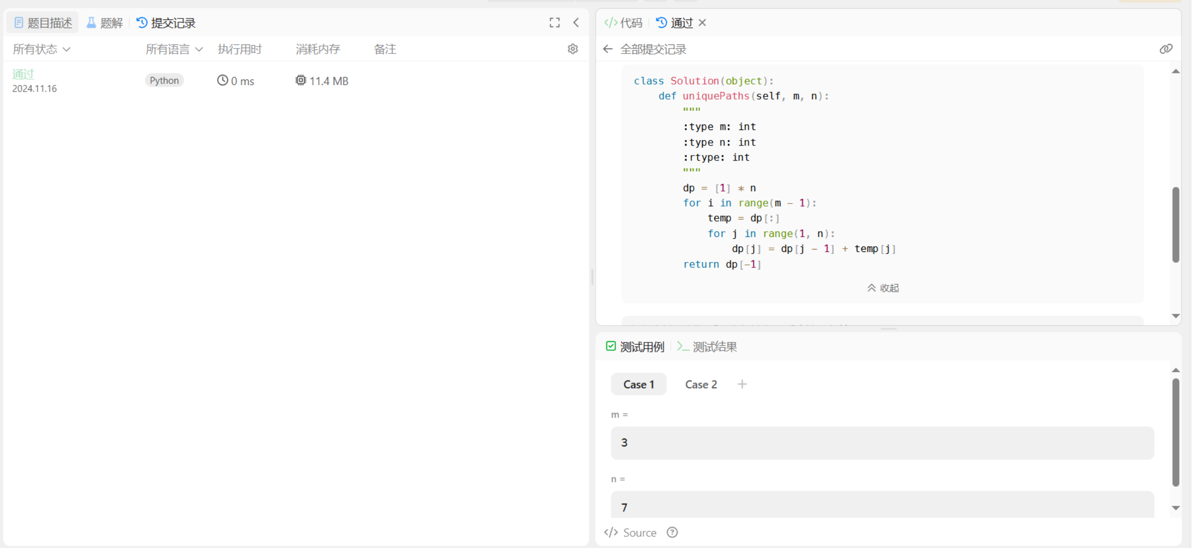
dp, https://leetcode.cn/problems/unique-paths/

思路：暴力，当然也可以直接用组合数，没什么区别，毕竟计算组合数也需要时间，***因此时间复杂度其实是一样的。***

代码：

```python
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        dp = [1] * n
        for i in range(m - 1):
            temp = dp[:]
            for j in range(1, n):
                dp[j] = dp[j - 1] + temp[j]
        return dp[-1]
```

代码运行截图 （至少包含有"Accepted"）



所有状态	所有语言	执行用时	消耗内存	备注
通过 2024.11.16	Python	0 ms	11.4 MB	

```python
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        dp = [1] * n
        for i in range(m - 1):
            temp = dp[:]
            for j in range(1, n):
                dp[j] = dp[j - 1] + temp[j]
        return dp[-1]
```

Case 1 Case 2 +

m =

3

n =

7

sy358: 受到祝福的平方

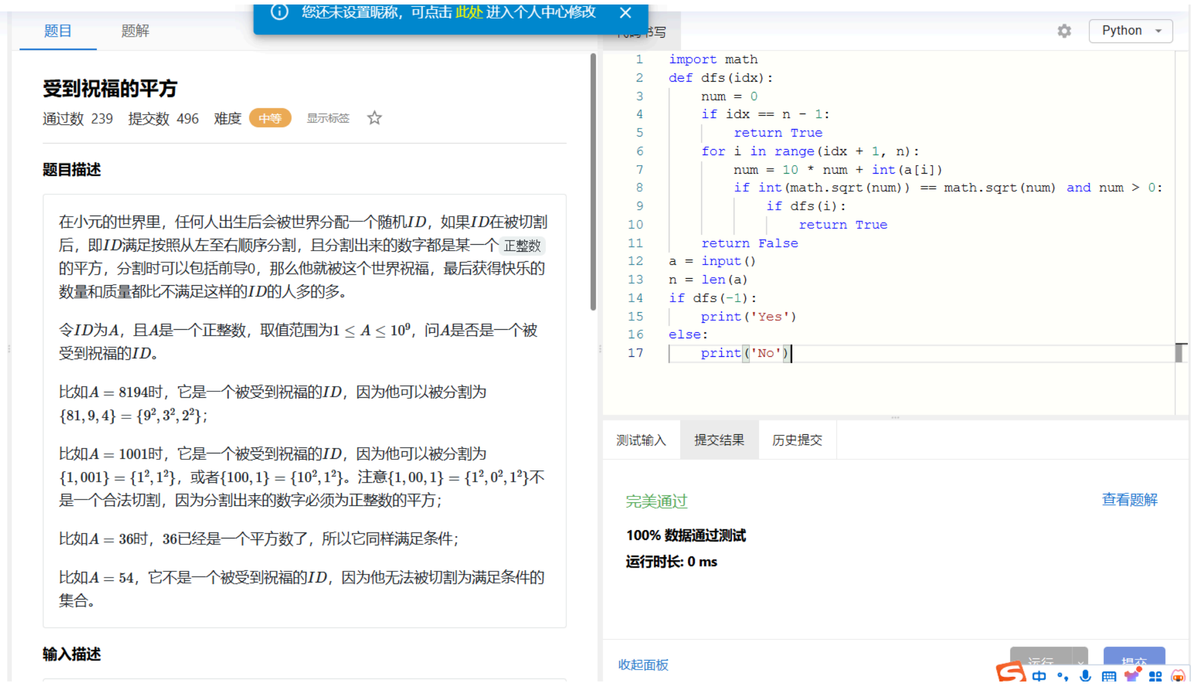
dfs, dp, https://sunnywhy.com/sfbj/8/3/539

思路:

代码:

```python
import math
def dfs(idx):
    num = 0
    if idx == n - 1:
        return True
    for i in range(idx + 1, n):
        num = 10 * num + int(a[i])
        if int(math.sqrt(num)) == math.sqrt(num) and num > 0:
            if dfs(i):
                return True
    return False
a = input()
n = len(a)
if dfs(-1):
    print('Yes')
else:
    print('No')
```

代码运行截图 （至少包含有"Accepted"）



受到祝福的平方

通过数 239 提交数 496 难度 中等

题目描述

在小元的世界里，任何人出生后会被世界分配一个随机ID，如果ID在被切割后，即ID满足按照从左至右顺序分割，且分割出来的数字都是某一个 正整数 的平方，分割时可以包括前导0，那么他就被这个世界祝福，最后获得快乐的数量和质量都比不满足这样的ID的人多的多。

令ID为A，且A是一个正整数，取值范围为$1 \le A \le 10^9$，问A是否是一个被受到祝福的ID。

比如$A = 8194$时，它是一个被受到祝福的ID，因为他可以被分割为$\{81, 9, 4\} = \{9^2, 3^2, 2^2\}$;

比如$A = 1001$时，它是一个被受到祝福的ID，因为他可以被分割为$\{1, 001\} = \{1^2, 1^2\}$，或者$\{100, 1\} = \{10^2, 1^2\}$。注意$\{1, 00, 1\} = \{1^2, 0^2, 1^2\}$不是一个合法切割，因为分割出来的数字必须为正整数的平方;

比如$A = 36$时，36已经是一个平方数了，所以它同样满足条件;

比如$A = 54$，它不是一个被受到祝福的ID，因为他无法被切割为满足条件的集合。

输入描述

```python
import math
def dfs(idx):
    num = 0
    if idx == n - 1:
        return True
    for i in range(idx + 1, n):
        num = 10 * num + int(a[i])
        if int(math.sqrt(num)) == math.sqrt(num) and num > 0:
            if dfs(i):
                return True
    return False
a = input()
n = len(a)
if dfs(-1):
    print('Yes')
else:
    print('No')
```

完美通过

100% 数据通过测试

运行时长: 0 ms

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

本次作业题相当简单，没有什么思维难度，全是模版题，重点是如何简化代码。自己正在leetcode上面做题，也在做老师的讲义上面的题目