

# Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

4 апр 2024, 02:44:38  
старт: 28 окт 2023, 21:14:17  
финиш: 26 ноя 2023, 03:14:16  
длительность: 28д. 5ч.  
...

Объявления жюри

Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

Положение участников Задачи Посылки

## 39. Путь спелеолога

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Пещера представлена кубом, разбитым на N частей по каждому измерению (то есть на  $N^3$  кубических клеток). Каждая клетка может быть или пустой, или полностью заполненной камнем. Исходя из положения спелеолога в пещере, требуется найти, какое минимальное количество перемещений по клеткам ему требуется, чтобы выбраться на поверхность. Переходить из клетки в клетку можно, только если они обе свободны и имеют общую грань.

### Формат ввода

В первой строке содержится число N ( $1 \leq N \leq 30$ ). Далее следует N блоков. Блок состоит из пустой строки и N строк по N символов: # - обозначает клетку, заполненную камнями, точка - свободную клетку. Начальное положение спелеолога обозначено заглавной буквой S. Первый блок представляет верхний уровень пещеры, достижение любой свободной его клетки означает выход на поверхность. Выход на поверхность всегда возможен.

### Формат вывода

Вывести одно число - длину пути до поверхности.

### Пример

Ввод	Вывод
3	6
### ### .##  .#. .#S .#.	
### ... ###	

### Примечания

Нужно спуститься на уровень вниз, сделать два движения на запад, подняться на уровень вверх, сделать движение на юг, подняться на уровень вверх.

Язык Python 3.12.1

Набрать здесь Отправить файл

```
1 from collections import deque
2
3 fin = open('input.txt')
4 N = M = K = int(fin.readline())
5
6 cave = [[[0 for _ in range(K)] for _ in range(M)] for _ in range(N)] # Или наоборот?
7 start = (None, None, None)
8 # i = z, j = y, k = x
9 # z = 0 >>> Вышли!
10 for i in range(N):
11     fin.readline()
12
13     for j in range(N):
14         for k, point in enumerate([x for x in fin.readline().rstrip()]):
15             if point == '#':
16                 cave[i][j][k] = 0
17             elif point == '.':
18                 cave[i][j][k] = 1
19             elif point == 'S':
20                 cave[i][j][k] = 1
21                 start = (i, j, k)
22
23 def getNextPos(z_pos, y_pos, x_pos, N, M, K):
24     delta = (1, 0, 0), (-1, 0, 0), (0, 1, 0), (0, -1, 0), (0, 0, 1), (0, 0, -1)
25
26     res = []
27     for step in delta:
28         next_z = z_pos+step[0]
29         next_y = y_pos+step[1]
30         next_x = x_pos+step[2]
31
32         if (0 <= next_z) and (next_z <= N-1) and (0 <= next_y) and (next_y <= M-1) and (0
33             res.append((next_z, next_y, next_x))
34
35     return res
36
37 def getMinPath(cave, start):
38 <
```

Отправить осталось 100 попыток

Предыдущая Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
30 окт 2023, 03:13:25	95157330	39	Python 3.12.1	OK	-	107ms	4.59Mb	-	-



1. Гистограмма



2. Красивая строка



3. Коллекционер Диего



4. Контрольная работа



5. Хорошая строка



6. Операционные системы lite



7. SNTP



8. Минимальный прямоугольник



9. Сумма в прямоугольнике



10. Скучная лекция



11. Стек с защитой от ошибок



12. Правильная скобочная последовательность



13. Постфиксная запись



14. Сортировка вагонов lite



15. Великое Лайнландское переселение



16. Очередь с защитой от ошибок



17. Игра в пьяницу



18. Дек с защитой от ошибок



19. Хипуй



20. Пирамидальная сортировка



21. Три единицы подряд

22. Кузнечик

23. Калькулятор

24. Покупка билетов

25. Гвоздики

26. Самый дешевый путь

27. Вывести маршрут максимальной стоимости

28. Ход конём

29. Кафе

30. НОП с восстановлением ответа



31. Поиск в глубину



32. Компоненты связности



33. Списывание



34. Топологическая сортировка



35. Поиск цикла



36. Длина кратчайшего пути



37. Путь в графе



38. Блохи



39. Путь спелеолога



40. Метро