

ShallowsExplanation

Every time you walk a Lane, there must be a path, then the maximum waterline of this path is the minimum depth of all Lanes in the path

Then we create a new Path object, which stores the next Lane in Path, and the maximum waterline depth of the current Path. In the algorithm, we will continue to update these two values.

We use a priority queue queue to store Path objects, and arrange them in reverse order based on the depth of Path. The queue will start with the shallowest path each time, then the path we take each time is the largest depth among all current paths.

Initially, we converted the Lanes connected to the source into Path objects and put them into the queue, and then continuously took the Path objects from the queue through the while loop.

Lane.arrive and Path.depth in the current Path object are the maximum depth of the arrival port, and then we create a Path for all Lanes of the current arrival.

The depth of the new Path is the minimum of the depth of the current Path and Lane.depth , and then put it into the queue

The solution to the problem can be obtained until all ports are traversed or the queue is empty.

ZiHan Zhang (23194782)

Xi He (22872431)

```
ShallowsTester x
00_example: PASS
01_singleton: PASS
02_pair: PASS
03_split: PASS
10_medium: PASS
11_medium: PASS
50_small_random: PASS
51_small_random: PASS
52_small_random: PASS
90_random: PASS
91_random: PASS
92_random: PASS
All tests passed
01_large:
00_300k_random: PASS
01_300k_random: PASS
02_300k_random: PASS
All tests passed
All tests passed
```