awice ★ 4269 Last Edit: September 26, 2018 11:15 PM 2.3K VIEWS

31

We can use Dijkstra's algorithm to find the shortest distance from the ball to the hole. If you are unfamiliar with this algorithm, how it works is that we process events in priority order, where the priority is (distance, path_string). When an event is processed, it adds neighboring nodes with respective distance. To repeatedly find the highest priority node to process, we use a heap (priority queue or 'pq'), where we can add nodes with logarithmic time complexity, and maintains the invariant that pq[0] is always the smallest (highest priority.) When we reach the hole for the first time (if we do), we are guaranteed to have the right answer in terms of having the shortest distance and the lexicographically smallest path-string.

When we look for the neighbors of a location in the matrix, we simulate walking up/left/right/down as long as we are inside the bounds of the matrix and the path is clear. If during this simulation we reach the hole prematurely, we should also stop. If after searching with our algorithm it is the case that we never reached the hole, then the task is impossible.

```
def findShortestWay(self, A, ball, hole):
    ball, hole = tuple(ball), tuple(hole)
    R, C = len(A), len(A[0])

    def neighbors(r, c):
        for dr, dc, di in [(-1, 0, 'u'), (0, 1, 'r'),
                           (0, -1, 'l'), (1, 0, 'd')]:
            cr, cc, dist = r, c, 0
            while (0 <= cr + dr < R and
                    0 <= cc + dc < C and
                    not A[cr+dr][cc+dc]):
                cr += dr
                cc += dc
                dist += 1
                if (cr, cc) == hole:
                    break
            yield (cr, cc), di, dist

    pq = [(0, '', ball)]
    seen = set()
    while pq:
        dist, path, node = heapq.heappop(pq)
        if node in seen: continue
        if node == hole: return path
        seen.add(node)
        for nei, di, nei_dist in neighbors(*node):
            heapq.heappush(pq, (dist+nei_dist, path+di, nei) )

    return "impossible"
```

Comments: 7 Best | Most Votes | Newest to Oldest | Oldest to Newest

Type comment here... (Markdown is supported)

Post

salamanderrex ★ 32 December 25, 2018 11:46 AM

Is it possible to use A* Search to make this even faster?

2 Reply

jedihy ★ 1160 February 1, 2017 5:00 AM

Awesome solution! It'd be better if no variable renaming.

1 Reply

feng-zhe ★ 77 September 17, 2018 5:15 AM

Same idea. But your code is really nice!

0 Reply

rksys ★ 2 February 22, 2018 6:49 PM

your code is so beautiful!

0 Reply

livelearn ★ 159 April 17, 2017 7:23 AM