2020CS10315

Akarsh Jain

Lab 1 Grp 2

Assignment 6.

Prob 1, 2 in .py file.

Prob 3

1. in .py file.

2. Explanation of Algorithm

- input file is read and stored as list of lists.
- Starting and ending point ('S' and 'E') are found out.
- A list s is maintained which contains the order in which points are visited in the maze. The last element of s being the current position in the maze.
- A array is maintained which is at starting a copy of the maze. But then as we move, the previous position in the array is marked as 'X' so as to avoid circular paths.

- Now here a case arrises when we keep moving forward and marking previous position 'X', when the current location is surrounded in all directions by 'X' then in this case, we have to realise that this is not a viable path. Hence we pop the stack and place a 'X' at the current position. And Hence we now go to some other alternative path from the previous location. (If alternate not exist, we do the same thing for previous location, then previous - previous location and so on).

- Clearly, In this strategy we are not moving in circles and are ~~correctly~~ effectively handling cases when we reach dead end. This maze will not be solved if the end is not reachable.

- At the end if we encounter 'E' at current posn we terminate our loop. good format gives the list s in a 'U', 'D', 'R', L' format and problem solved.

Correctness of goodformat(s)

Invariant: elements before index $i-1$ are converted to good form.

Base case: $i=1$. But there are no element before index 0. Hence correct.

Maintenance: Let it be true for $i=n$. Then since we are checking for $i-1=n-1$ whether, $s[i-1]=(a,b)$, $s[i]=(c,d)$

if $a=c$ then clearly row is same

if $d=b+1$ the clearly right movement

else left.

if $a=c+1$ then up

if $a+1=c$ then down.

Hence, it will hold for $i=n+1$ also.

Termination: $i=len(s)-1$. Then clearly after execution of loop only last element will be left which is popped since it given posn of 'E' which is useless. Hence desired output is obtained.

Correctness of move function is trivially established from explanation of code and why is it implemented.

Correctness of traverseMaze

1st for loop -- since this operates for all strings in the variable b. Removes the '\n' and splits string with sep(" ").
Hence correctness is trivial.

Similarly 2nd loop runs for all $0 \le i < b$ and $0 \le j < a$. Hence, it covers all the elements in the matrix M. Hence, if there exist ~~start~~ 's' and 'E' it must find the location.

Proof for 3rd loop by cases.

Intention: If we find that element at current position is 'E' then loop is stopped which is what is the while condition.

If current is not 'E' then we can either do one of these to find 'E' -

1. move up
2. move left
3. move down
4. move right

— only if there is '_' or 'E' at the prospective posn.

5. ~~go~~ go to prev location if all sides are covered by 'X' and chose other path from there by marking current posn as 'X' and poping list ~~a~~ s

Also as we move we keep marking prev posn 'X' so that we do not go in circles

Now since all cases are dealt and constraints

Namely, ① no circles

② shortest path not needed

③ only U, D, R, L moves

are also satisfied.

Hence the function and the loop both are correct.

(correctness of overall func. can be directly inferred from the explanation.)