Q 2.1)

**Subproblems:** for each and , Let be the problem of determining , the maximum number of possible unique paths to reach from with only two kinds of moves (down one cell or right one cell).

**Recurrence:** for and ,

At any point that interpreting no possible paths through that cell.

There are only two possible ways to reach any cell which is either from its top cell or its left cell since only moving down one cell or right one cell are allowed. So, the number of possible unique paths to reach any cell is the sum of possible unique paths of its top cell and its left cell.

Since depends on (its top cell) and (its left cell), we solve subproblems in increasing order of then .

**Base cases:**

If or then

Interpreting no possible paths to outside the whole cells.

Interpreting only one possible path to the start which is staying at the start unless there is a box.

If , it means the warehouse layout meets the requirement for having a way to reach the exit with only two kinds of moves. The warehouse layout doesn’t meet the requirement otherwise.

Overall time complexity is for iterating through all elements in a 2D array of size (all elements from one column then move to another column). And each of subproblems is solved in constant time.

Q 2.2)

**Subproblems:** for each and , Let be the problem of determining , the minimum number of boxes that must be removed in order to reach from with only two kinds of moves (down one cell or right one cell).

**Recurrence:** for and ,

At any point that interpreting no boxes need to be removed to make a path to that cell.

There are only two possible ways to reach any cell which is either from its top cell or its left cell since only moving down one cell or right one cell are allowed.

The number of boxes that must be removed in order to reach any cell is either from its top or left cell then increase it by 1 if the cell has a box on it (since that box must be removed in order to reach the cell).

So, the smallest number could be found by choosing the minimum number of boxes that must be removed in order to reach its top and its left cell (among two of them).

Since depends on (its top cell) and (its left cell), we solve subproblems in increasing order of then .

**Base cases:**

If or then

Interpreting no possible paths to outside the whole cells no matters how many boxes are removed.

.

Interpreting a box must be removed if it happens to be at the starting cell.

is the smallest number of boxes that must be removed to meet the requirement.

Overall time complexity is for iterating through all elements in a 2D array of size (all elements from one column then move to another column).

And each of subproblems is solved in constant time.

Q 2.3)

Since no matter what unique paths we take, we would need to move down one cell ( to ) then move to the right one cell ( to ) at least once (not in a strict order) creating a corner ( to ) in a path, in order to reach the exit at the bottom-right from top-left (since ).

A shortcut is when we move down one cell and to the right one cell at the same time ( to ). That means whenever we have a corner, we would be able to use a shortcut instead because we would end up in the same destination location (and as mentioned above that there’s always at least one corner in a path).

So, if there exists a path to the exit, there would have at least one corner that could be using a shortcut.

Q 2.4)

**Subproblems:** for each and , Let be the problem of determining , the minimum sum of hazard ratings of a path from to taking exactly one shortcut with only two kinds of moves, and , the minimum sum of hazard ratings of the path (without taking shortcuts if possible). , the maximum difference if a shortcut is taken along the path comparing with not taking any, and , the minimum number of shortcuts that is necessary to make a possible path to .

**Recurrence:** for and ,

**Base cases:** if or then and . .

If or then , and .

If then and .

If then and .

The minimum sum of hazard ratings of a path to the exit taking exactly one shortcut is ( couldn’t be undefined since the warehouse passed the safety inspection).

At any point that has a box, and would all be undefined since it is not possible to be at that point making it impossible to have sum of hazard ratings.

, minimum sum of hazard ratings at any point, is from the minimum of its top and left points (since only two kinds of moves are allowed). However, if both points have boxes, then a shortcut is the only option (so take it).

, maximum difference if a shortcut is taken along the path, is from the maximum difference of previous shortcut or minimum of at top and left points (interpreting taking a new shortcut at top or left corner if it could reduce sum of hazard more than the old shortcut). Since a short cut could be taken only once throughout any paths, if a shortcut is necessary taken (2 boxes case), then no more shortcut should be taken after that. However, if another shortcut must be taken, then this is undefined and the path is invalid (not a possible path to the exit with only one shortcut).

, minimum sum of hazard ratings taking one shortcut, is from the minimum of sum of hazard of 3 possible ways to the point (top, top-left(corner) and left) subtracting by their maximum difference if a shortcut is taken (except from the corner, that is assuming to take a new shortcut to the exit).

Overall time complexity is for iterating through all elements in a 2D array of size (all elements from one column then move to another column).