APP1: Be Amazed Report

Members:

- DO Son Tung
- SOSNIN Andrey
- MAJDEDDINE ALHAFEZ
- Eman AL-SHAOUR
- Antoine COLOMBIER

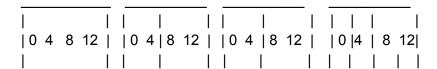
Maze Generation

Problem:

- Input: 2 dimensions p,q as height and width.
- Output: A tree which represents different steps of subdivision. Leaves correspond to elementary 1x1 cells of the maze. Two nodes are siblings iff each of them has a cell (an entry or an exit point, depending on the point of view) from which we can move (in the maze) to the other one in one move. Each node contains a reference to its parent and offspring, if any. Each non-leaf node contains the position of the door which connects its offspring (see below). Each leaf contains its coordinate. There is no imposed order on the nodes at this stage.

Solution:

We number individual cells on the grid top to bottom, left to right from 0 to p*q - 1 where p and q are vertical and horizontal dimensions, e.g. for n = p*q = 16:



Every time a split is made we save the position of the gateway represented by a pair of cells adjacent to it (e.g. {7,11}, {5,6}, {13,14}...).

One can obtain cartesian coordinates of a cell by performing a euclidean division by p. It is also easy to express the condition that exit E should be on the outer wall: $E \setminus p == 0 \mid E \setminus p == q-1 \mid E \setminus p == 0 \mid E \setminus p == q-1 \mid E \mid p == q-1 \mid E \mid$

Solving the maze

Input:

The coordinate of the exit cell
The coordinate of the starting cell

General outline(precise procedures will be detailed later):

Descend from root, ordering necessary branches of the tree as you go. The process is recursive: we start by considering the starting and the exit cells; then, at each meaningful subdivide we order the sons and we apply anew the method on the left, then on the right son, substituting one of the points of interest (i.e. start or exit) by the position of the gate of that subdivide. (A subdivide is 'meaningful' when the points of interest are in different sons)

```
find_path(node, start, exit)

if node is leaf:

add node to the path
if node == exit:

return

if start == exit: # non-essential check
add start to the path
return
else:
```

determining position of a cell wrt to a gateway:

(the comparisons work only because cell is always in one of the rooms separated by {a,b})

Notes (not to include in the report):

we consider 1x1 rather than 1xn elementary cell

we do not consider that nodes are ordered wrt to the exit cell after generation because I don not see how to do it with random exit without additional operations;

Numbering from 0 to n-1 rather than using (x,y) coordinates barely serves any purpose, but it's significantly easier to draw.

Complexity Analysis

Generation:

The generation algorithm is a "divider and conquer" method. The algorithm method is called twice; one for every branches of the node:

Best case: The tree is balanced; every sub-areas located on same parent node own the same dimensions.

 $T(n) = 2T(\frac{n}{2}) + C$, therefore T(n) = O(n) by Master theorem

Worst case: Random subdivision leads to the tree being unbalanced; one of the sub-areas will have a smallest area, while the other will have the maximum, and this on every levels. However, the asymptotic complexity remains the same because the tree with n leaves has at most 2n leaves.

Note: as well as during the solving part of the problem (as we will see), this algorithm creates a stack of recursive calls equal (up to an additive constant) to the height of the tree.

Path finding:

In the worst case, we have to go through every cell (for example: if the maze have the spiral shape), so the total solution time must be at least n. On the other hand, the proposed solution pass through each node at most once. So we take at most 2n nodes therefore the complexity is O(n)