**Description of puzzle algorithm**

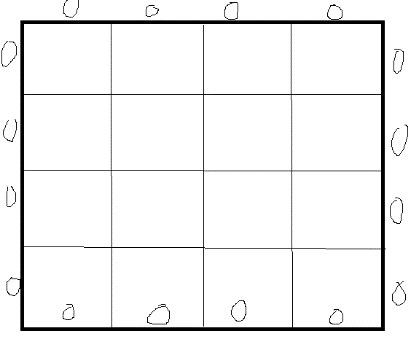
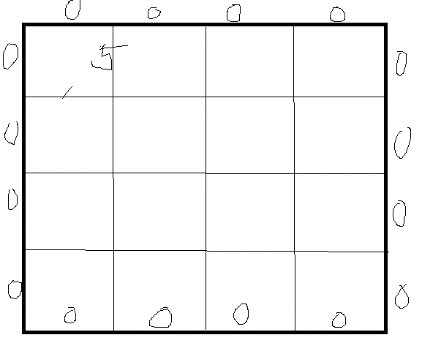
Author: Vadym

createdAt: 12/06/2023

**Util class( I will explain with 4 \* 4 .txt file)**

1. **General Algorithm**

First of all, I know the border, values of all borders are 0,

-> 

**First, find the candidates of 0th position, then take one of them, ( on my screen, I choosed [0,0,5,1]**

**Second, find the candidates of 1th position, then take one of them,**

**(candidates should have this value, [5, 0, x, x], I can easily find that )**

**…**

**Until 15th position, I can take the candidates of each position.**

This is the general idea of this program.

In my program , in the main () function, run

**public static void main() {**

**…**

**Util.selectCandidate(0);**

**}**

**©** Util.selectCandidate () function is a kind of recursive function.

It calls it byitself in it for search other candidate until return true.

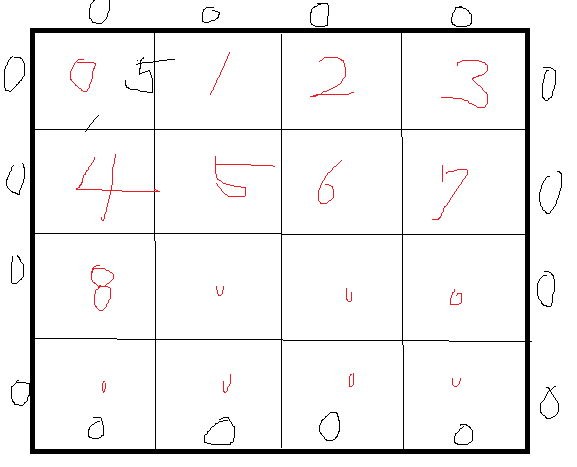
findCandidatePuzzles(int index):

as you can see on the figure 1, findCandidatePuzzle function returns the candidates for the index position. In my case on the figure, left = 0, top = 0,

then if index == 1 -> left = 5, top = 0,

and index == 4 -> left = 0, top = 1, …

index = 15 => I can know the left , top, right and bottom.



This is the main functions on my algorithm