


Algorithmics	Student information	Date	Number of session
	UO:288764	15/4/24	6
	Surname: Sampedro	 Escuela de Ingeniería Informática Universidad de Oviedo	
	Name: Carlos		



## Activity 1. Numeric Square. Description of the algorithm

### NumericSquare Class

First, I use the Buffer Reader to read the file. Then, I create a matrix with all the values and in the pair rows I add a space so it is shown correctly and once it is distributed accordingly, I process all Row and Column Operations for the given matrix.

### NumericSquareSolver Class


First, with the use of GetRowOperations() or getRowOperations() I get all the row or col operations in a String [] list.

Having that, I get all values in an operation and if they are equal to '?' I substitute them by -1 so that I can get them all in an Integer List. This is done in getValues(String operation).

I also have a method for getting the expectedResult and the list of operators for a given operation.

After getting all those values I use them in the method getResults(operation= where I make use of the recursive method GetAllValidCombinations() for a given set of values, that takes all values that are equal to -1 ('?') from 0 to 9 using backtracking, gets the result of the proposed values, and if they are a valid solution for the given operation of a row or col they return them.

The previous method provides with all the valid solutions for each row or column of the matrix, now it is merged together in getRowOrColResults(String [] operations), now with the use of backtracking all possible matrixes are created from that data in generateAllPossibleMatrixes() It returns from row0 sol0, one adding row1 sol0, and row2 sol0, then row0 sol0, row1 sol1, row2 sol0, next with rown2 sol1 and so on...

Algorithmics	Student information	Date	Number of session
	UO:288764	15/4/24	6
	Surname: Sampedro	 Escuela de Ingeniería Informática Universidad de Oviedo	
	Name: Carlos		



Finally, we've got to check which of those matrixes give a valid value, given that for a 3x3, row0, row1 and row2 values in index 0 make up the column operation number 0, those values should satisfy the columnOperation[0] expected result, this is checked in isValidMatrix(), where it checks if each row and column satisfy that condition.

By iterating all possible matrixes and checking if they are valid, we get the solutions to the NumericSquare problem.

## Activity 2. Complexity and measurements

I think that the complexity of my algorithm was higher than the expected one.

I tried once I was doing the matrix combinations to check whether they were valid or not, by passing the colOperations[] as a parameter, and if they were not valid then returning a null and calling the Java Garbage collector, but I was not able to make it work properly.

This algorithm functions correctly until test05, from then it throws OutOfMemoryError.

```
Also this was implemented if(b==0 || a%b!=0 ) {
    return 0;
}
```

As 4/3 f.e was not suitable for getting an answer, provided a mail sent by Vicente to one student, because it gives decimals.

The complexity of my code is first for Calculating all possible values from 0 to 9, it would be  $O(9 - 0)^{(\text{number of ?})}$ . Then for calculating the possible valid matrixes it would be  $O(\text{row})^{\text{number of solutions}}$ , and for checking their validity  $O(n^2)$

Algorithmics	Student information	Date	Number of session
	UO:288764	15/4/24	6
	Surname: Sampedro		
	Name: Carlos		

## Measurements

Test case	Time for first solution(ms)	Time for all solutions(ms)	Number of solutions found
Test00	2(LoR)	7(Lor)	1
Test01	25(LoR)	64	12
Test02	2(LoR)	1(Lor)	1
Test03	317	202	3
Test04	642	658	2
Test05	1964	2973	5
Test06	OutOfMemoryError	OutOfMemoryError	-
Test07	OutOfMemoryError	OutOfMemoryError	-