

PTAI Assignment 3

Students : Praneeth Jakkaraju, Tudor Chiribes, Laura Gonzalez

Student Id's : 19239985, 19240205, 19240029

List of Tasks attempted :

- 9565186b.json - Praneeth
- 28bf18c6.json - Tudor
- ea786f4a.json - Laura

Github Link to Solution :

<https://github.com/techdem/ARC/tree/master/src>

Task Description :

Task 9565186b.json :

Pattern:

The pattern in this task is to identify the cells which are colored same in $n*n$ grid and get all maximum cells colored with a particular color. Consider these cells in the input grid and color all the other cells with grey which makes our output.

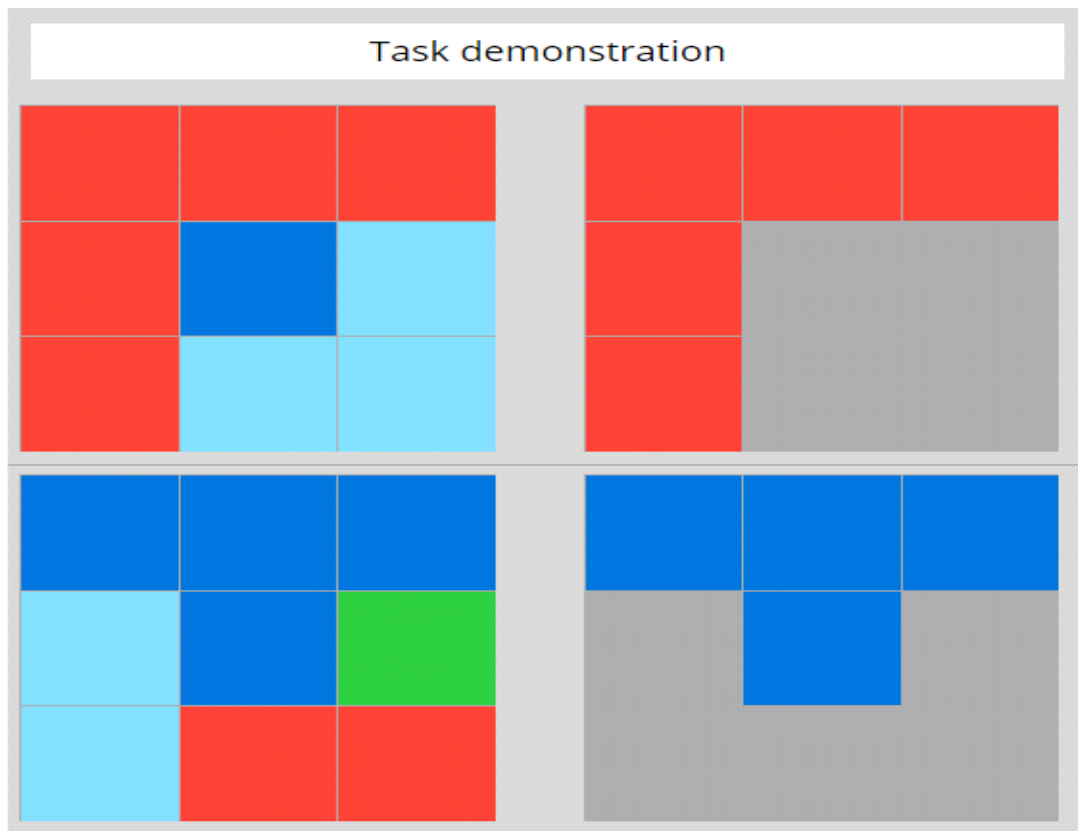
Solution – Solve Method

Input is passed as python list of lists to the solve method. First step in solve method is to convert the input to numpy array. The coding part is made generic to work for any kind of input color code. To get all the color codes in input, numpy unique function is used which returns color code and count of cells which have that particular color. Then I map it to a dictionary using zip function, and get a dictionary of color code and count of cells which have that color. Then I get the max count which implies that the cells which have this color shouldn't be altered and assign it to a variable. Then I loop through the input array and compare each element of array with the unaltered code to make these cells unaltered and alter all the other cells with value 5(grey). At the end the function returns the input array with altered cells.

This solution works for all the inputs and outputs given in input file.

Python Libraries:

As part of this solution, I have used numpy package. Which is also mentioned in requirements.txt as this package has to be installed before running the solution. Numpy is used to get the unique values from array, here we get unique colors in the grid with count of cells.



Task Description :

Task ea786f4a.json:

Pattern:

The pattern in this task is to identify the color in the center of the input grid and fill both of the diagonals with the same color and leave all of other cells as it is.

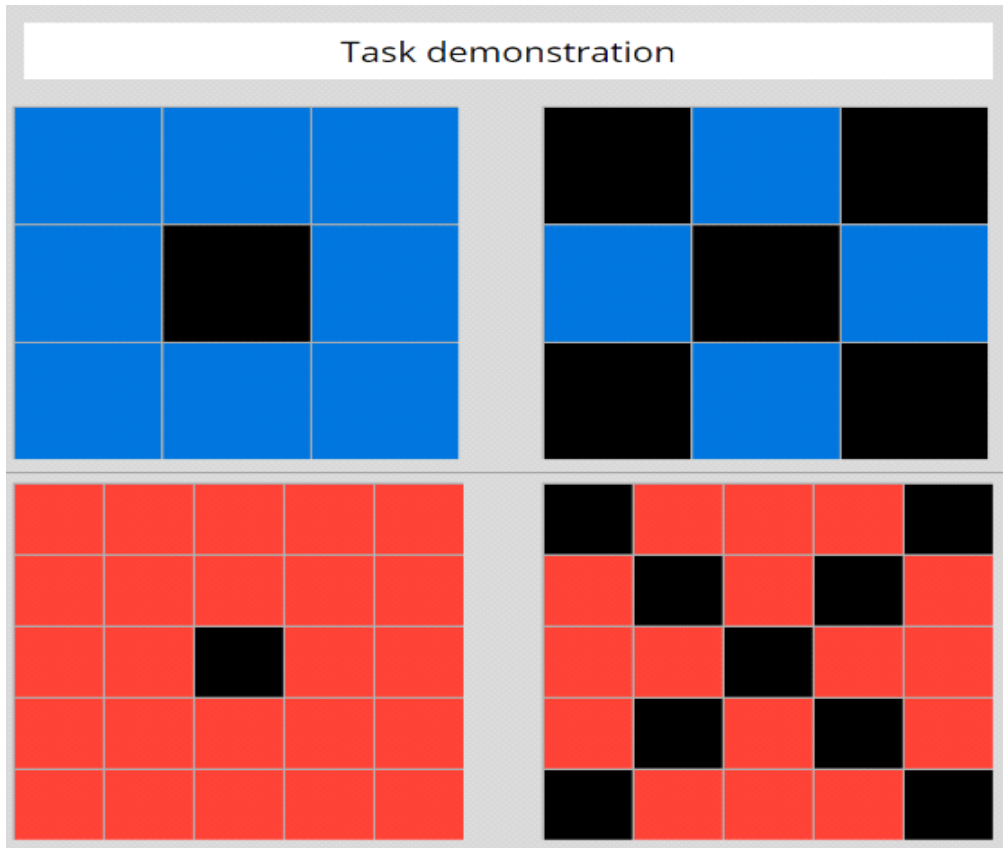
Solution – Solve Method

Input is passed as python list of lists to the solve method. First step in solve method is to convert the input to numpy array. From this array we get the color of the center cell in the diagonal. And then we fill the diagonals with the selected color using numpy diagonal method.

This solution works for all the inputs and outputs given in input file.

Python Libraries:

As part of this solution, numpy is used to fill the diagonals of the matrix with the selected color code.



Task Description :

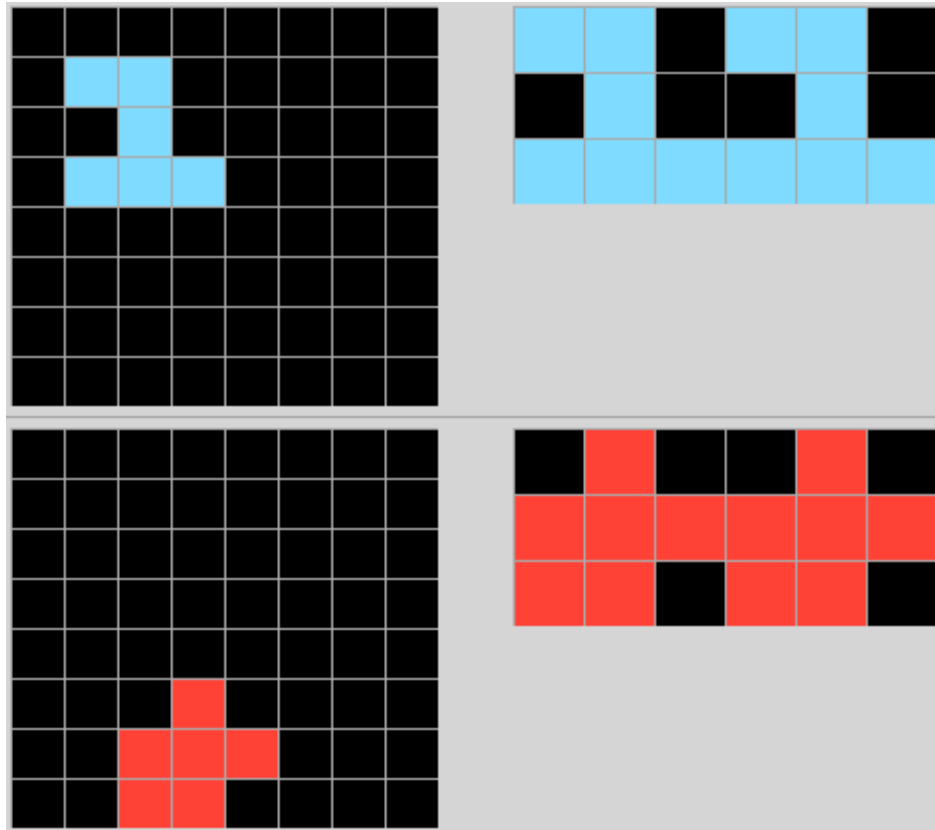
Task 28bf18c6.json:

Pattern:

The input is a grid of 8x8 from which a smaller grid of 3x3 contains non-negative values. The output is generated by combining two copies of the 3x3 array into a 3x6 one.

Solution

The solve method will receive a list of lists which can be represented as a 2D array with rows and columns. From this resulting 8x8 grid we can find the 3x3 grid with non-negative values by iterating over each possible 3x3 grid and calculating the one with the highest value. This is achieved by using a nested for loop with n^2 complexity. At each individual element we are adding the values of the other eight ones that form the grid. If for a grid the total value is higher than the stored largest value, each row is duplicated and the result is stored. At the end, a 3x6 list of lists with the duplicated grid is returned.



Measuring Intelligence : Intelligence is a relative term and can be defined differently by different persons. Though it is very difficult for machines to replicate Human behavior perfectly, this field is being evolved. Some of the key issues in AI systems are replicating human behavior, since we humans learn everything from experience and keep constantly evolving our selves with intelligence and environment. Its not about skill alone for any intelligent system to perform some tasks effectively, many factors such as practical experience, conditions and environment in which task has to performed contribute heavily. Skills again can be tied up closely with the task that we are about to solve, just like humans. Not all humans have the capability to do every task with same effectiveness, systems do have limited capability in terms of task diversity.

As part of solutions to some of these problems, our tasks were involved in making a psychometric intelligent model. This is a kind of domain, where its purely human intelligence and skills which can detect the patterns without much difficulty. Making system detect the patterns with experience which is achieved by training the model on training data and evaluating the performance by testing data. Though we dint use Machine learning models to train and achieve the patterns, we have selected few tasks where we coded the pattern for a specific task. These solutions derive the ability of the system to recognize the inputs and derive the output pattern, which is task specific currently. It can be evolved task agnostic by mapping the behavior of input and output of each task and training the model based on these mappings.

Summary: The tasks we chose vary in terms of patterns. As we chose all colored tasks, the commonality between these tasks were to identify the color and detect the size of input. Every task we had to identify

the color since the output was also of the same color, and in one of the tasks we had to identify the color count since the output depended on the color which was maximum filled. In terms of patterns too, the tasks differ with each other. There are two tasks in which we used Numpy, to use its features in detecting the shape and code a pattern. Then we had the similarity in identifying the input grid sizes. As none of our tasks has the output which contains empty cells, and the output is heavily tied to input size. One more common thing in our solutions was to print the output. To solve this we defined a common class with print code in it, and shared the same code all through out the three tasks.

Work Split Up: As mentioned in selected tasks, we took one task each. Everyone was responsible in delivering the code and understanding the purpose for their respective tasks. We connected with each other virtually and contributed to this report.