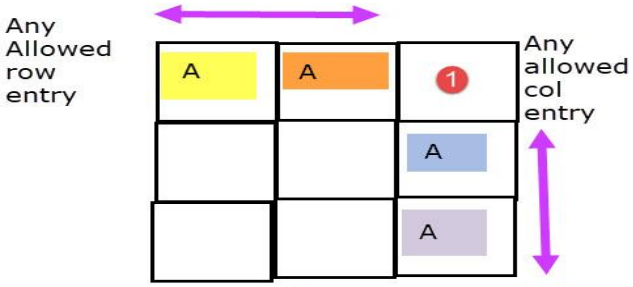
We can give it a head start up to here. Allowed column and allowed row

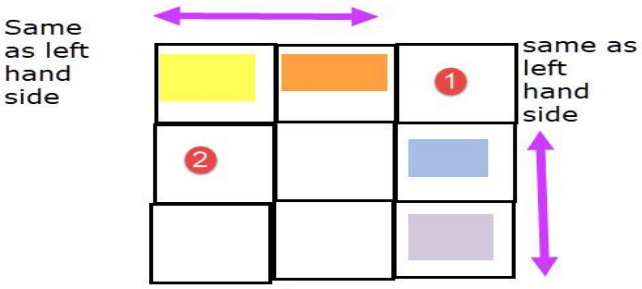


In order to insert 1 We would need to search through allowed row permutations

1

&& 1

We now continue from previous constraint and now need following:

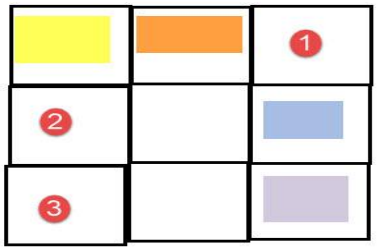


In order to insert 2 We would need to search through allowed row and col permutations

2

&& 2

We now continue from previous constraint and now need following:



In order to insert 3 We would need to search through allowed row and col permutations

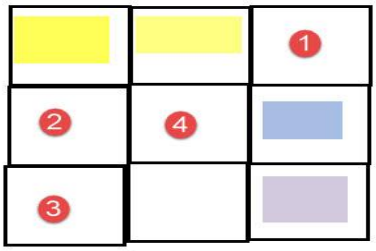
2 ALREADY OK

&& 2 3

&& 3

&& 3

We now continue from previous constraint and now need following:



In order to insert 4 We would need to search through allowed row and col permutations

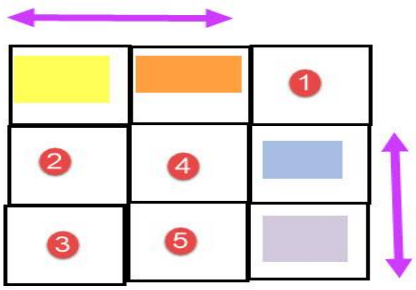
4 ALREADY OK

2 4

2 ALREADY OK

4

We now continue from previous constraint and now need following:



In order to insert 5 We would need to search through allowed row and col permutations

5

5 3

3 ALREADY OK

5 4

5

WE WOULD NEED TO OVERCOME 11 CONSTRAINTS SHOULD WE HAVE ALL ALLOWING PAIRS AT OUR DISPOSAL AND CHOOSE TO POPULATE 4 GRIDS WHICH DO NOT INTERFERE.

I WOULD THEN ADD ADDITIONAL 5 GRIDS BASED ON AVAILABILITY OF ALL CONSTRAINTS WITHIN AVAILABILITY ARRAY.

It might require all 65.8billlion unordered pairs. All my current storage would be breached