Solution@11-08-2023

2570. merge two 2d arrays by summing values

```
class Solution {
  public int[][] mergeArrays(int[][] nums1, int[][] nums2) {
    final int kMax = 1000;
    List<int[]> ans = new ArrayList<>();
    int[] count = new int[kMax + 1];

    addCount(nums1, count);
    addCount(nums2, count);

    for (int i = 1; i <= kMax; ++i)
      if (count[i] > 0)
        ans.add(new int[] {i, count[i]});

    return ans.stream().toArray(int[][] ::new);
  }

  private void addCount(int[][] nums, int[] count) {
    for (int[] idAndVal : nums) {
      final int id = idAndVal[0];
      final int val = idAndVal[1];
      count[id] += val;
    }
  }
}
```

662. Maximum Width of Binary Tree

```
class Solution {
    public int widthOfBinaryTree(TreeNode root) {
```

```
        if(root == null) return 0;

        Deque<TreeNode> dq = new LinkedList<>();
        dq.add(root);

        int max = 1;
        while(!dq.isEmpty()) {
            while(!dq.isEmpty() && dq.getFirst() == null )    dq.removeFirst();
            while(!dq.isEmpty() && dq.getLast() == null )    dq.removeLast();
            max = Math.max(max, dq.size());
            int size = dq.size();
            for(int i =0; i <size; i++) {
                TreeNode temp = dq.poll();
                if(temp == null) {
                    dq.add(null);
                    dq.add(null);
                } else {
                    dq.add(temp.left);
                    dq.add(temp.right);
                }

            }
        }
        return max;
    }
}
```