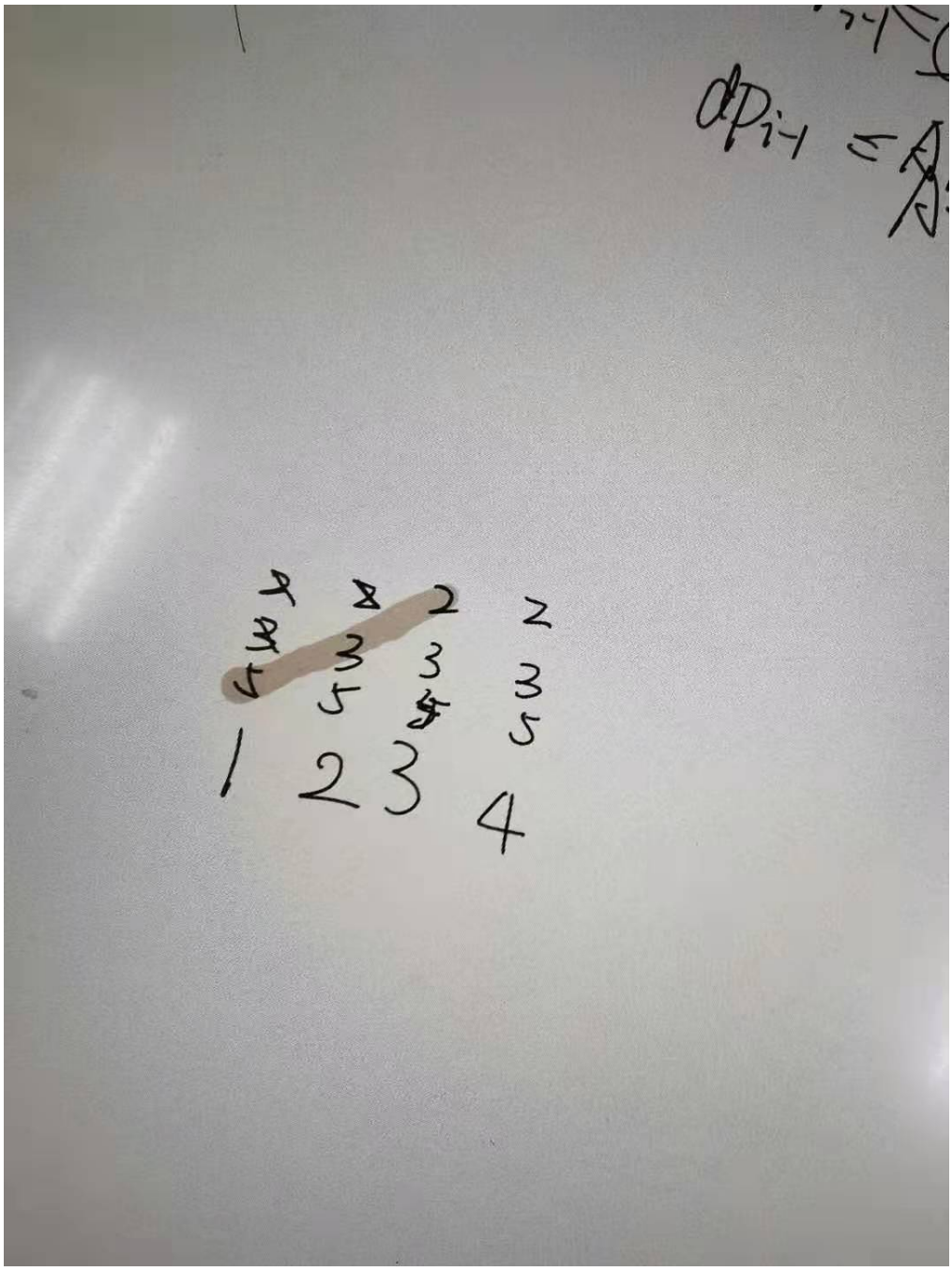
丑数2

- 这个题目我开始以为根据n的总数在指数的位置处按顺序填入数字就可以了，但实际上存在2*2小于5的情况，前者用来两位，后者只用了一位，所以我想的这种方法只适用于5，7，11这种，多用一个指数的值肯定比少用一个指数的值要大的情况，而且这种方法会产生重复元素，又需要进行指数运算，显然不可取
- 那么只能直接通过数的值进行递推公式生成后续的序列了，做法根据题解有两种（这里我们忽略暴力求解法）
    - 通过哈希集合来去除重复元素，通过最小堆来进行排序，这种想法借助STL能够很快想出来
    - 通过动态规划和数组指针的方法
        - 这个方法跟我想过的一个结构题很类似，每个元素都有*2 *3 *5三个机会，用完一次之后，就要减少一次，最后一个元素取决与前面的具有剩余机会的所有元素
        - 这里反映到指针上就出现了一个有趣的现象，跟题解一样的现象
        - 

- 题解如下:

```java
class Solution {
    public int nthUglyNumber(int n) {
//a,b,c按顺序填就好了，主要是探究a,b,c的填法
        if(n==1){
            return 1;
        }
        int[] dp = new int[n+1];
        dp[1]=1;
        int num2 = 2;
        int num3 = 3;
        int num5 = 5;
        int dp2 = 1;
        int dp3 = 1;
        int dp5 = 1;
        for(int i=2;i<n+1;i++){
            num2 = dp[dp2]*2;
            num3 = dp[dp3]*3;
            num5 = dp[dp5]*5;
            int val = Math.min(Math.min(num2,num3),num5);
            dp[i]=val;
            System.out.println(val);
            if(val == num2){
                dp2+=1;
            }
            if(val == num3){
                dp3+=1;
            }
            if(val == num5){
                dp5+=1;
            }
        }
        return dp[n];
    }
}
```

- 千万要注意这里的三个if，这里没有用if - else的结构，因为存在2* 3 和3* 2相等的情况，通过这个结构可以让两个指针同时移动，不然队列中会出现重复的元素!!!