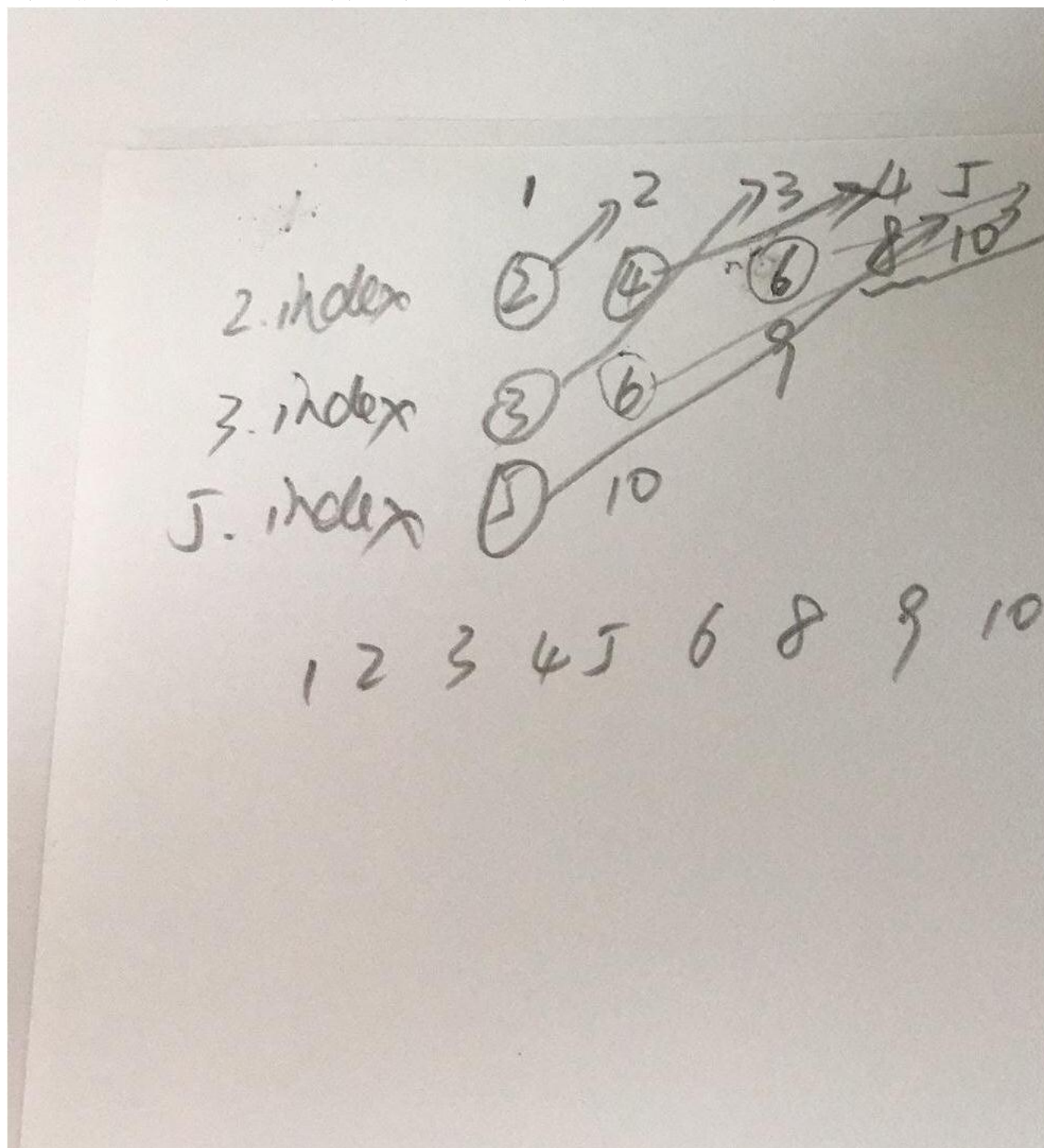


使用一个数组将之前已经求出来的丑数保存起来，然后依次用2, 3, 5乘上数组中已经求出来的丑数，选出其中最小的一个保存在数组中。假设数组中存的最大丑数是max，那么只要求2, 3, 5乘以数组中的某一个数刚好比max大，然后选出三者间最小的就可以。如下图所示。



```

package com.bladeToOffer;
/*
 * 丑数
 */
public class B49 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int index=14;
        System.out.println(GetUglyNumber(index));
    }
    public static int GetUglyNumber(int index)
    {
        if(index==0)
            return 0;
        //如果index==0, 返回0
        int ugly[]=new int[index];
        ugly[0]=1;
        //当index!=0时将第一个丑数1放到数组中。
        int uglyindex=1,min;
        int nextindex2=0,nextindex3=0,nextindex5=0;
        while(uglyindex<index)
        {
            min=ugly[nextindex2]*2<ugly[nextindex3]*3?ugly[nextindex2]*2:ugly[nextindex3]*3;
            //min用于保存新产生的大于当前数组最大值的最小丑数
            min=min<ugly[nextindex5]*5?min:ugly[nextindex5]*5;
            ugly[uglyindex]=min;
            if(ugly[nextindex2]*2<=min)
                nextindex2++;
            //如果ugly[nextindex2]小于当前数组的最大值, nextindex2++;
            if(ugly[nextindex3]*3<=min)
                nextindex3++;
            //如果ugly[nextindex3]小于当前数组的最大值, nextindex3++;
            if(ugly[nextindex5]*5<=min)
                nextindex5++;
            //如果ugly[nextindex5]小于当前数组的最大值, nextindex5++;
            uglyindex++;
            //uglyindex++;寻找下一个丑数
        }
        return ugly[index-1];
    }
}

```