Rakuten Coding Exam

```
        N = len(nums)

        if N < 3:
            return 0

        if N == 3:
            return 1 if ((nums[0]-nums[1])==(nums[1]-nums[2])) else 0

        num_slices = i = pre = 0

        while (i + 2) <= (N-1):
            if (nums[i] - nums[i+1]) == (nums[i+1] - nums[i+2]):
                num_slices += (pre + 1)
                pre += 1
            else:
                pre = 0
            i += 1

        return
```

```
    def nthUglyNumber(self, n):
        dp=[1]*(n+1)
        p2=p3=p5=1
        for i in range(2,n+1):
            np2,np3,np5=2*dp[p2],3*dp[p3],5*dp[p5]
            minimum=min(np2,np3,np5)
            dp[i]=minimum
            if np2==minimum:
                p2+=1
            if np3==minimum:
                p3+=1
            if np5==minimum:
                p5+=1
        return dp[-1]
```

Problems given:

1. ugly nos
2. Velocity stability