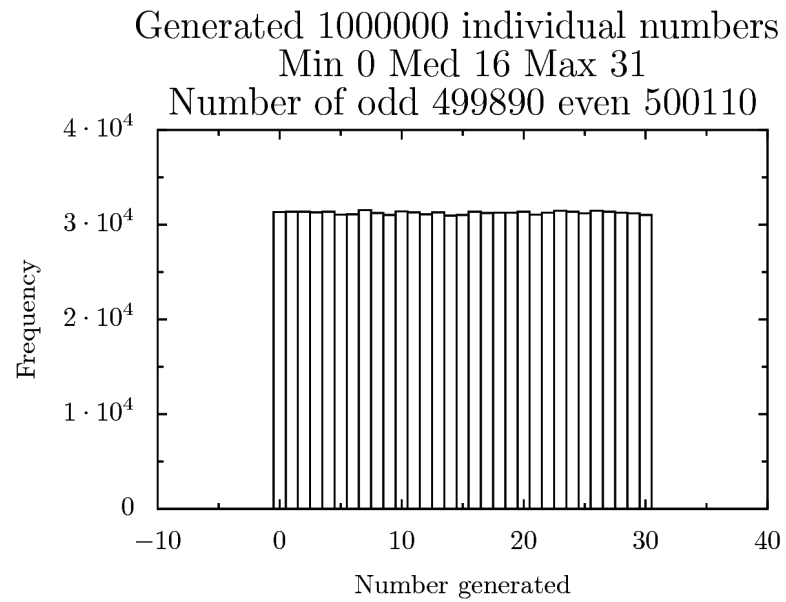


This function was used to generate the following data integers. I only created numbers between 0 and 31.

```
def smallInt():  
    randfunc = os.urandom  
  
    number = ord(randfunc(1)) >> (3)  
    return number
```



This is a modified form of the scheme used by pycrypto to generate a N-bit number from the random bytes returned by the CSPRNG.

```
def getRandom(N):  
    """  
    Return an random number between 0 and N-bits in length.  
    """  
    randfunc = os.urandom  
  
    S = randfunc(N/8)  
  
    odd_bits = N % 8  
    if odd_bits != 0:  
        char = ord(randfunc(1)) >> (8-odd_bits)  
        S = chr(char) + S  
  
    value = 0L  
    length = len(S)  
  
    for i in range(0, length):  
        value = (value << 8)  
        value = value + ord(S[i])  
  
    return value
```

Generated 100000 individual numbers 0-10 bits long

Number of odd numbers 49993

Number of even numbers 50007

Min 0

Med 509

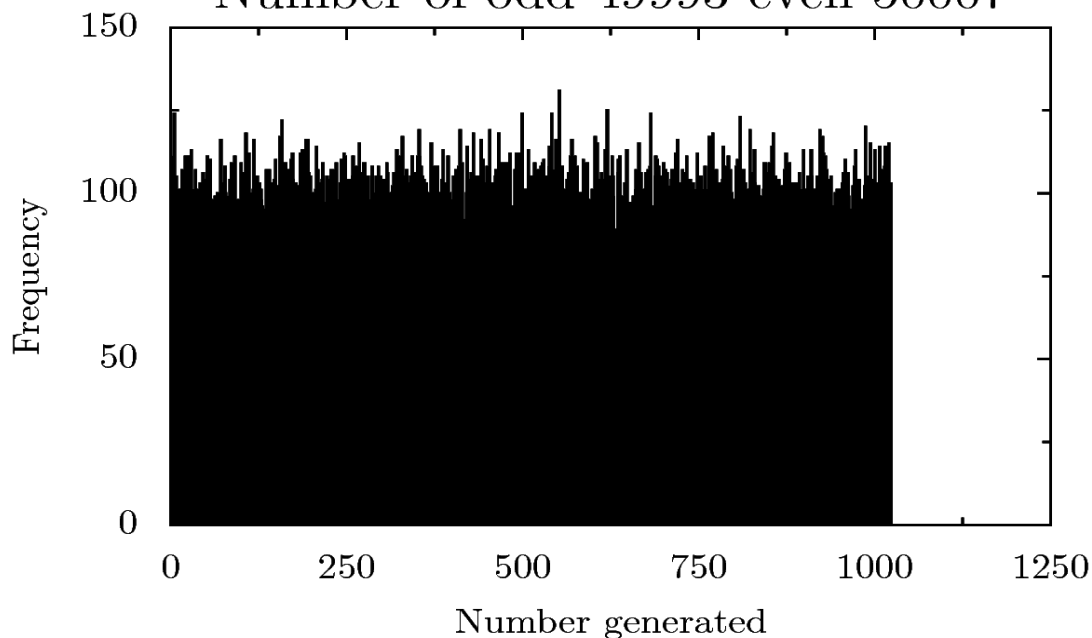
Max 1023

The maximum frequency was 131 for number 551

Generated 100000 individual numbers

Min 0 Med 509 Max 1023

Number of odd 49993 even 50007



Generated 1000000 individual numbers 14 bits long
Number of odd numbers 500642
Number of even numbers 499358
Min 0
Med 8195
Max 16383
The maximum frequency was 101 for number 7687

