In [1]:
```
pip install mmh3
```

```
Collecting mmh3
  Downloading mmh3-3.1.0-cp39-cp39-win_amd64.whl (15 kB)
Installing collected packages: mmh3
Successfully installed mmh3-3.1.0
Note: you may need to restart the kernel to use updated packages.
```

Task1

In [3]:
```
import mmh3
from bitarray import bitarray
import random

r1=random.randint(0,100)

r2=random.randint(0,100)
while(r2==r1):
    r2=random.randint(0,100)

r3=random.randint(0,100)
while(r3==r1 or r3==r2):
    r3=random.randint(0,100)

bitarr= bitarray(64)
bitarr.setall(0)

class BloomFilter:

    def __init__(self):
        return

    def Hash1(self,a):
        return (mmh3.hash(a,r1))%64

    def Hash2(self,a):
        return (mmh3.hash(a,r2))%64

    def Hash3(self,a):
        return (mmh3.hash(a,r3))%64

    def insert(self,a):

        i1=self.Hash1(a)
        i2=self.Hash2(a)
        i3=self.Hash3(a)

        bitarr[i1]=1
        bitarr[i2]=1
        bitarr[i3]=1

    def check(self,a):

        i1=self.Hash1(a)
        i2=self.Hash2(a)
        i3=self.Hash3(a)
```

```
        if(bitarr[i1]==1 and bitarr[i2]==1 and bitarr[i3]==1):
            print("The item {} is probably present".format(a))
        else:
            print("The item {} is definitely not present".format(a))

        return None


bf=BloomFilter()

s= 'Hello'

bf.insert(s)
bf.insert('World')


bf.check(s)
bf.check('World')
bf.check('world')

#print(bitarr)
```

```
The item Hello is probably present
The item World is probably present
The item world is definitely not present
```

Task 2

In [4]:

```
import random

count = 10000

list_data = [int(random.random()*count) for _ in range(count)]

def trailing_0(hash_value):

    if(hash_value==0):
        return 0

    else:

        bin_value=bin(hash_value)
        count=0
        l=len(bin_value)
        ind=l-1
        bv=bin_value[ind]

        while(bv!='b' and bv!='1'):
            count=count+1
            ind=ind-1
            bv=bin_value[ind]

        return count

def flajolet_martin(list_data):

    tail_0_max = 0
```

```
    for data in list_data:
        hash_value = hash(data)
        tail_0_max = max(tail_0_max, trailing_0(hash_value))

    return 2**tail_0_max

estimated_count = flajolet_martin(list_data)

print ('estimated_count: {}'.format(estimated_count))

#print(list_data[0])
#print(hash(list_data[0]))
```

```
estimated_count: 4096
```

In [ ]: