

▼ 68_Adnan Shaikh

```

import random
import math
from statistics import median, mean

class Flajolet_Martin:

    def __init__(self,k=3,l=5,L=32,elements=[]):
        self.k = k
        self.l = l
        self.L = L
        self.total_hm = self.k*self.l
        self.hms = []
        self.elements = elements
        self.add_hms()

    def generate_hm(self,a=None, b=None):
        if not a:
            a = random.randrange(1,2**self.L-1,2)
        if not b:
            b = random.randrange(1,2**self.L-1)
        return lambda x: (a*x+b)%(2**self.L)

    def add_hms(self):

        for _ in range(self.k):
            self.hms.append([self.generate_hm() for x in range(self.l)])

    def trailing_zeros(self,x):
        return (x&-x).bit_length() - 1

    def cal_distinct_elements(self,ele = []):
        if not ele:
            ele = self.elements

        avg_hms = []
        for gp in self.hms:
            hm_vals = []
            for hm in gp:
                trail = 0
                for x in ele:
                    trail = max(trail, self.trailing_zeros(hm(x)))
                hm_vals.append(2**trail/0.77351)
            avg_hms.append(mean(hm_vals))

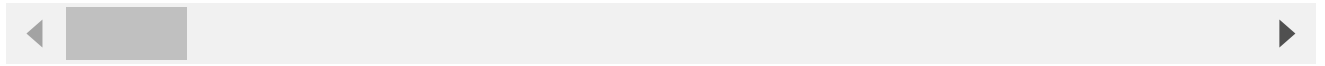
        return round(median(avg_hms))

stream_ele = random.choices(range(0,10000),k=10000)

```

```
for x in stream_ele:  
    print(x,end=" ")
```

3781 149 1482 5862 9349 6232 115 7851 6749 7558 7996 6604 4558 3050 7768 2622



```
fm = Flajolet_Martin(k=25,l=100,elements=stream_ele,L=14)
```

```
print(f"Flajolet Martin Approximation: {fm.cal_distinct_elements()}\n\  
Actual No. of Distinct Elements: {len(set(stream_ele))}")
```

Flajolet Martin Approximation: 6791
Actual No. of Distinct Elements: 6326

[Colab paid products](#) - [Cancel contracts here](#)

✓ 22s completed at 17:22

