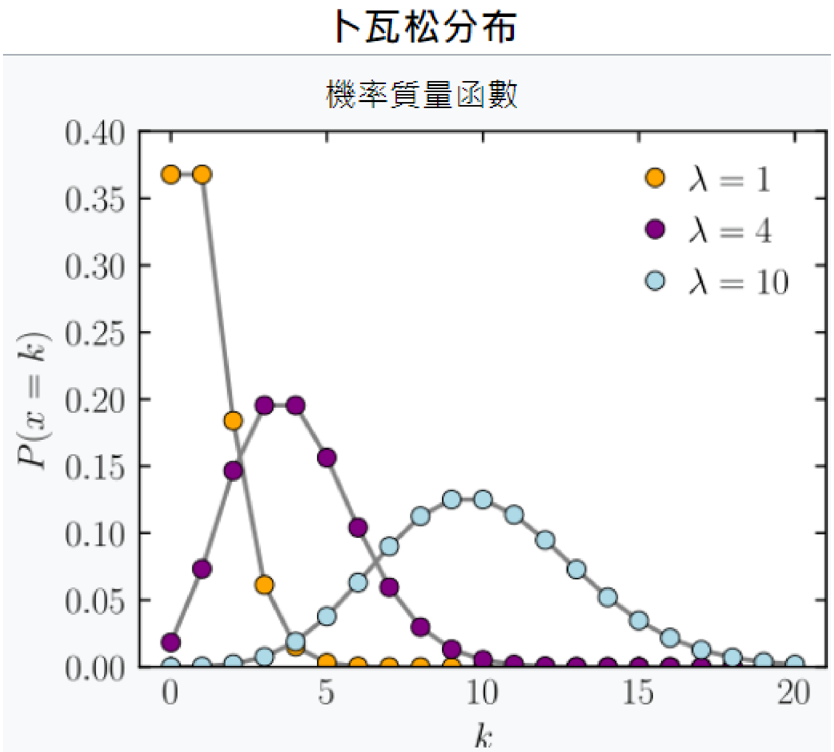
學號：N26111871 姓名：廖威任



上圖為 Poisson distribution 的機率質量函數，本次作業將聚焦在 method 1 及 method 2 是否基於 Poisson distribution 產生相同的分布。(圖片資料來源：維基百科)

```
mean = 10 #設定mean,可改為1、4等數值
interval_num = 35 #設定區間數
```

首先將設定 mean(λ)及區間數，經過測試後發現通常不會超出第 30 個區間，因此將區間個數設為 35。

- Method 1

```python
interval = [0] * (interval_num) #初始各區間的頭尾為0
sum = 0 #用來計算當前累計的區間
for i in range(interval_num): #計算各個區間
    p = math.exp(-1 * mean) * math.pow(mean, i) / math.factorial(i) #帶入公式求出p
    interval[i] = p + sum
    sum += p
```

利用公式： $p_i = P\{X = i\} = e^{-\lambda}\frac{\lambda^i}{i!} \quad i = 0, 1, \ldots$ 來求出各個區間的頭尾。

```python
count_1 = [0] * interval_num #統計各個區間的數量
for i in range(10000000): #利用隨機亂數產生一千萬筆資料
    u = random.random()
    for j in range(interval_num): #將隨機亂數和區間做比對
        if  u < interval[j]: #若符合條件則代表找到該區間
            count_1[j] += 1
            break
for i in range(interval_num):
    print("p",i,"：",count_1[i])
```

利用隨機變數產生一千萬筆資料，並和區間來做比對以找到符合的區間，最後統計出各個區間的分布。

λ=1

p 0	：	3679139
p 1	：	3677783
p 2	：	1840064
p 3	：	612836
p 4	：	153411
p 5	：	30698
p 6	：	5230
p 7	：	736
p 8	：	91
p 9	：	11
p 10	：	1
p 11	：	0
p 12	：	0
p 13	：	0
p 14	：	0
p 15	：	0
p 16	：	0
p 17	：	0

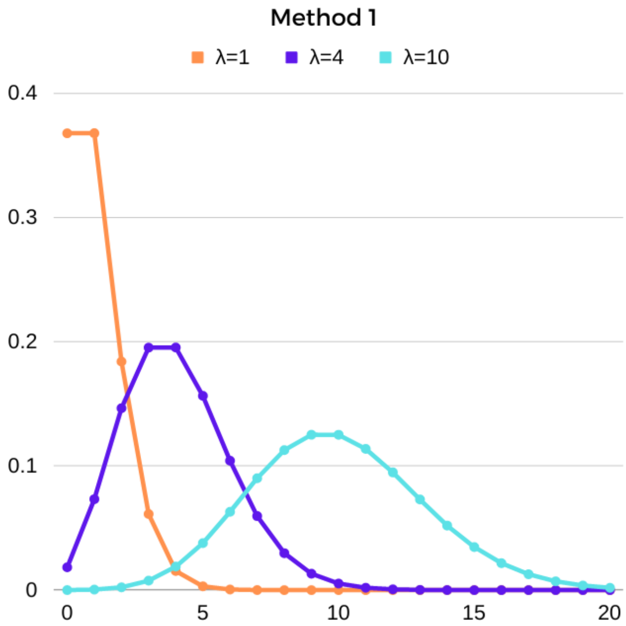
λ=4

p 0	：	183777
p 1	：	733001
p 2	：	1464372
p 3	：	1954687
p 4	：	1951824
p 5	：	1562819
p 6	：	1042182
p 7	：	595544
p 8	：	298057
p 9	：	132410
p 10	：	52618
p 11	：	19503
p 12	：	6521
p 13	：	1924
p 14	：	560
p 15	：	161
p 16	：	34
p 17	：	6

λ=10

p 0	：	457	p 18	：	70932
p 1	：	4499	p 19	：	37369
p 2	：	22721	p 20	：	18327
p 3	：	75581	p 21	：	8837
p 4	：	188603	p 22	：	4051
p 5	：	377740	p 23	：	1786
p 6	：	630959	p 24	：	741
p 7	：	902043	p 25	：	273
p 8	：	1125910	p 26	：	116
p 9	：	1250163	p 27	：	27
p 10	：	1251166	p 28	：	19
p 11	：	1136056	p 29	：	4
p 12	：	948837	p 30	：	2
p 13	：	730335	p 31	：	1
p 14	：	519854	p 32	：	1
p 15	：	347366	p 33	：	0
p 16	：	217647	p 34	：	0
p 17	：	127577			

上圖為 Method 1 在 λ=1、4、10 時的模擬分布結果。



上圖為 Method 1 在 λ=1、4、10 時的模擬分布圖，可以發現呈現 Poisson distribution。

- Method 2

```
count_2 = [0] * interval_num #統計各個區間的數量
for i in range(10000000): #產生一千萬筆資料
    prod = random.random()
    j = 1 #計算目前乘了多少次prod
    while prod >= math.exp(-1 * mean): #帶入公式來求出j
        prod *= random.random()
        j += 1
    count_2[j - 1] += 1

for i in range(interval_num):
    print("p",i,"：",count_2[i])
```

利用隨機變數產生一千萬筆資料，並帶入公式：

$$N = \mathrm{Max}\left\{n\colon \sum_{i=1}^{n} -\frac{1}{\lambda}\log U_i \leqslant 1\right\} = \mathrm{Max}\left\{n\colon U_1 \cdots U_n \geqslant e^{-\lambda}\right\}$$

來求出各個區間的資料數量。

λ=1

```
p 0  : 3679193
p 1  : 3681769
p 2  : 1835526
p 3  : 613354
p 4  : 153760
p 5  : 30508
p 6  : 5066
p 7  : 708
p 8  : 98
p 9  : 17
p 10  : 1
p 11  : 0
p 12  : 0
p 13  : 0
p 14  : 0
p 15  : 0
p 16  : 0
p 17  : 0
```

λ=4

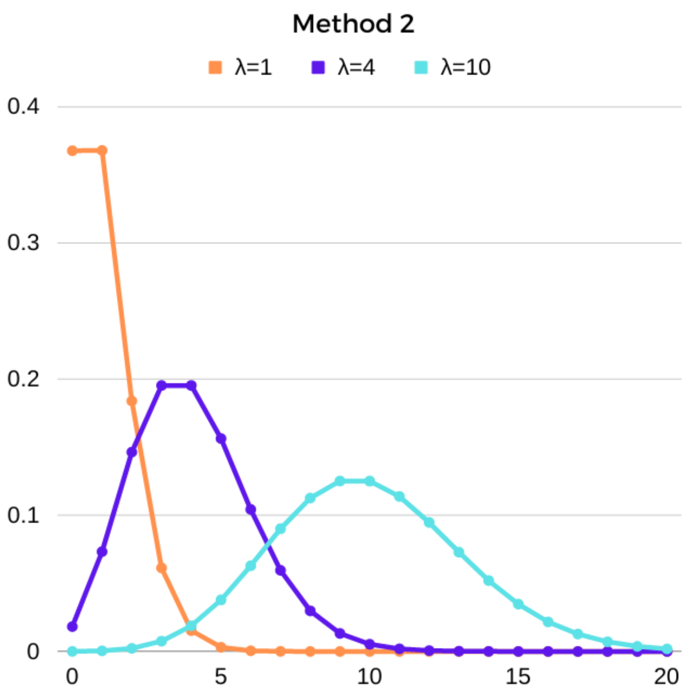
```
p 0  : 182808
p 1  : 731249
p 2  : 1466703
p 3  : 1953846
p 4  : 1952652
p 5  : 1563501
p 6  : 1042941
p 7  : 595220
p 8  : 297133
p 9  : 132194
p 10  : 53386
p 11  : 19112
p 12  : 6418
p 13  : 2028
p 14  : 602
p 15  : 155
p 16  : 38
p 17  : 10
p 18  : 4
```

λ=10

```
p 0  : 457
p 1  : 4532
p 2  : 22673
p 3  : 75941
p 4  : 189392
p 5  : 377877
p 6  : 631384
p 7  : 899321
p 8  : 1127237
p 9  : 1249941
p 10  : 1250208
p 11  : 1137371
p 12  : 947453
p 13  : 729975
p 14  : 520829
p 15  : 347100
p 16  : 217559
p 17  : 127960
```

```
p 18  : 71140
p 19  : 37080
p 20  : 18678
p 21  : 8954
p 22  : 4027
p 23  : 1752
p 24  : 711
p 25  : 280
p 26  : 99
p 27  : 44
p 28  : 17
p 29  : 5
p 30  : 3
p 31  : 0
p 32  : 0
p 33  : 0
p 34  : 0
```

上圖為 Method 2 在 λ=1、4、10 時的模擬分布結果。



上圖為 Method 2 在 λ=1、4、10 時的模擬分布圖，可以發現呈現 Poisson distribution。

- 結論

Method 1 及 Method 2 為基於 Poisson distribution 產生相同的分布，因為當 λ 值一樣時同樣要產生 Poisson random number 的話分部會類似。