

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np

from scipy.stats import chi2, expon, f, uniform
```

Вспомогательные методы

Метод генерации интенсивностей потоков. Возвращает n положительных интенсивностей, в сумме равных Λ .

In [2]:

```
def generate_lambdas(n, Lambda):
    lambdas = np.random.rand(n)
    lambdas -= (np.min(lambdas) - 1)
    lambdas *= Lambda / np.sum(lambdas)
    return lambdas
```

Метод генерации потока. Возвращает $size_i$ чисел, сгенерированных случайным образом из переданного в качестве параметра распределения $dist$. Можно получить поток из равномерного ($dist = \text{'uniform'}$) или из χ^2 ($dist = \text{'chi2'}$) распределений.

In [3]:

```
def generate_process(lambda_i, size_i, dist='uniform'):
    if dist == 'chi2':
        return chi2.rvs(1 / lambda_i, size=size_i)
    if dist == 'uniform':
        return uniform.rvs(scale=2/lambda_i, size=size_i)
```

Методы трансформации. Первый трансформирует список времён между событиями потока в список моментов времени, а второй трансформирует обратно.

In [4]:

```
def transform_times_to_moments(times):
    moments = list()
    moments.append(0)
    for i in range(0, len(times)):
        moments.append(moments[i] + times[i])
    return moments

def transform_moments_to_times(moments):
    times = list()
    for i in range(1, len(moments)):
        times.append(moments[i] - moments[i-1])
    return times
```

Метод отрисовки гистограммы суммарного потока и графика экспоненциального распределения с параметром Λ .

In [5]:

```
def draw_plot(summary_process, n, Lambda):
    edges = plt.hist(summary_process, range=(0, 6), bins=50, density=True)
    dots = np.linspace(np.min(edges[1]), np.max(edges[1]))
    expon_line = expon.pdf(dots, 0, 1 / Lambda)
    plt.plot(dots, expon_line, label='exp(Lambda)')
    plt.legend()
    plt.title(f'Количество потоков - n={n}')
    plt.show()
```

Основной метод

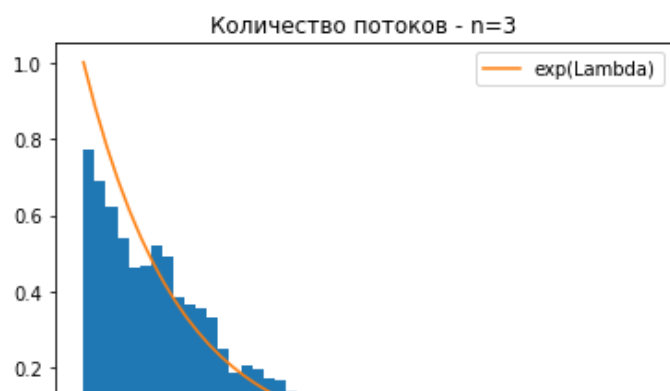
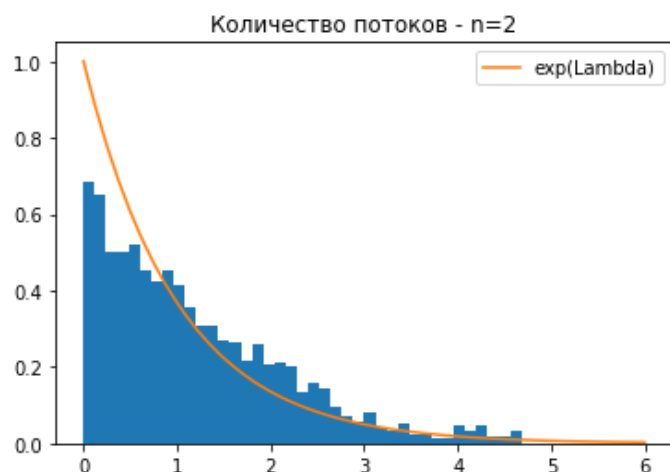
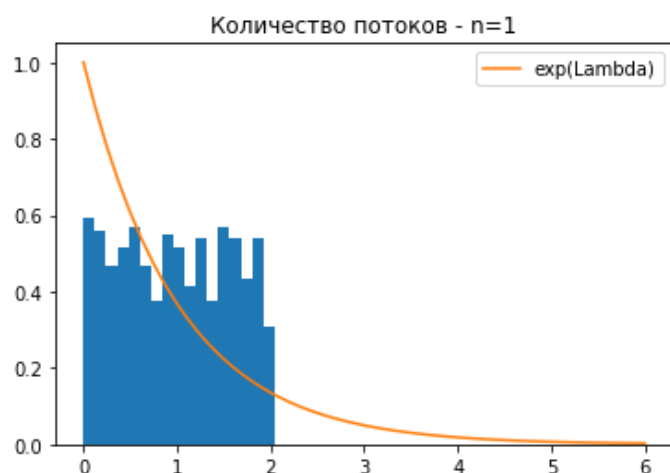
Для каждого $k \in \overline{1, n}$ генерируем k интенсивностей λ_i (в сумме равных Λ), после чего генерируем k потоков (из распределения $dist$), сохраняя каждый в общем списке моментов времени. Преобразуем суммарный список в список времён, строим гистограмму его распределения и сравниваем с распределением $Exp(\Lambda)$.

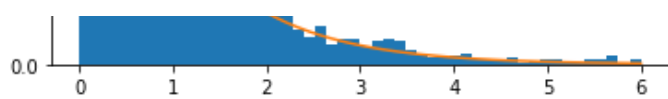
In [6]:

```
def main(n=10, Lambda=1, size_t=1000, dist='uniform'):  
    for k in range(1, n+1):  
        lambdas = generate_lambdas(k, Lambda)  
        summary_process = list()  
        for i in range(0, k):  
            summary_process += transform_times_to_moments(generate_process(lambdas[i], size_t, dist=dist))  
        summary_process.sort()  
        summary_process = transform_moments_to_times(summary_process)  
        draw_plot(summary_process, k, Lambda)
```

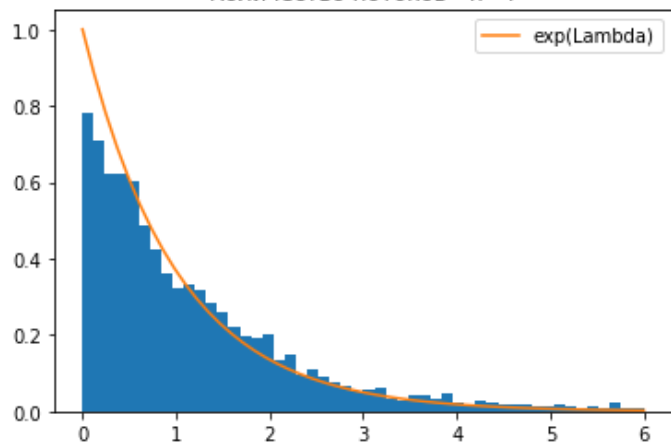
In [7]:

```
main()
```

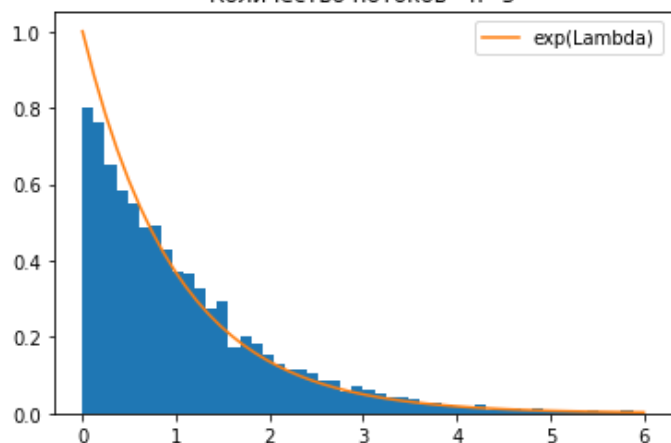




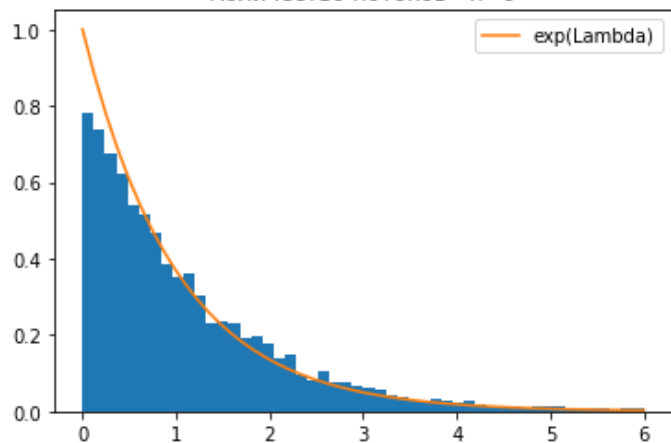
Количество потоков - $n=4$



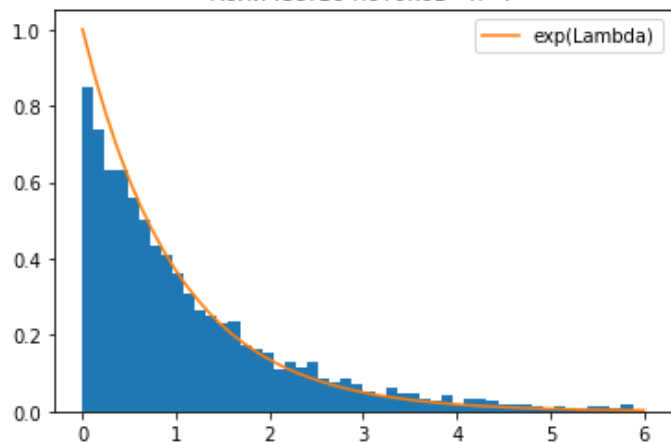
Количество потоков - $n=5$



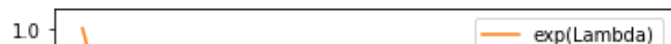
Количество потоков - $n=6$

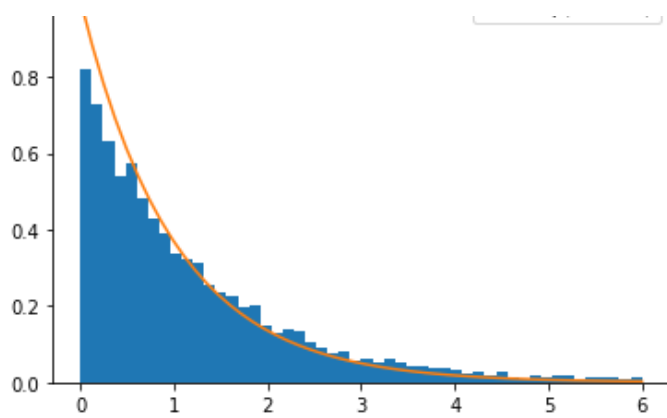


Количество потоков - $n=7$

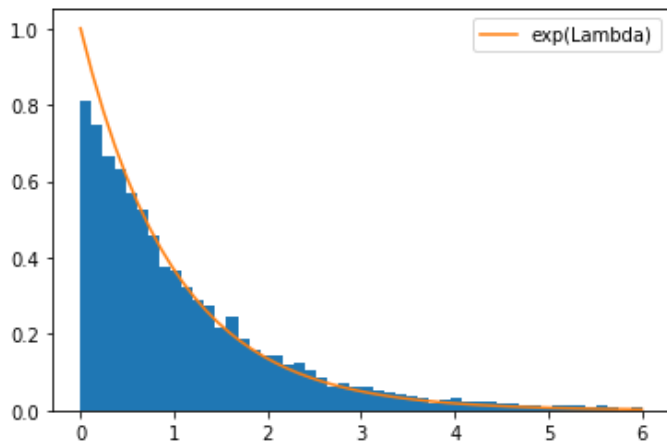


Количество потоков - $n=8$

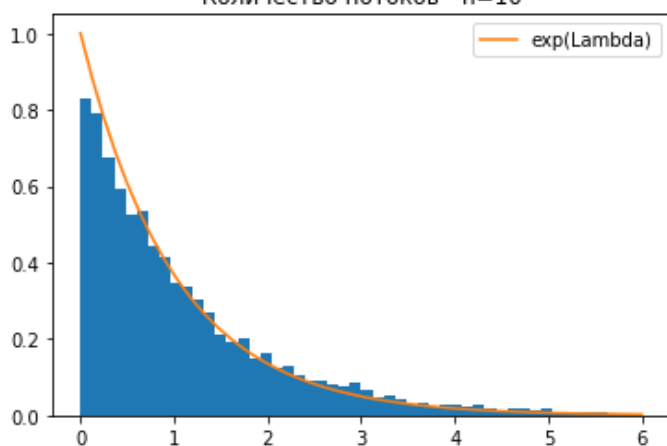




Количество потоков - n=9



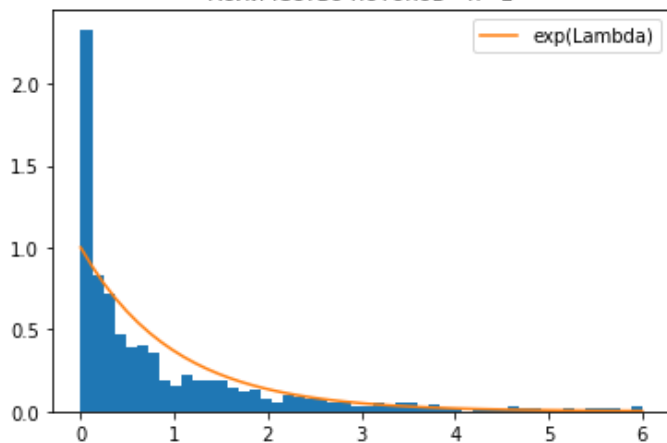
Количество потоков - n=10



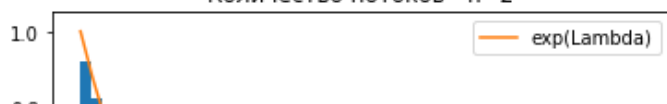
In [8]:

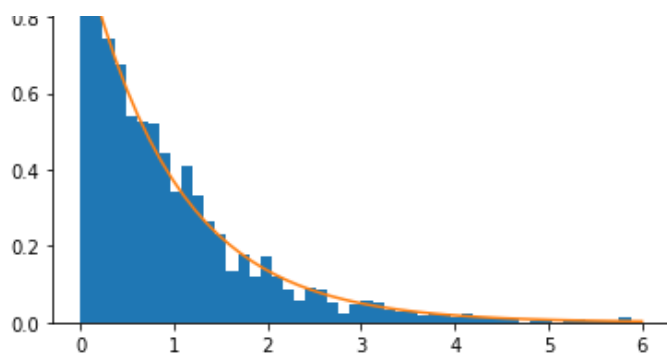
```
main(dist='chi2')
```

Количество потоков - n=1

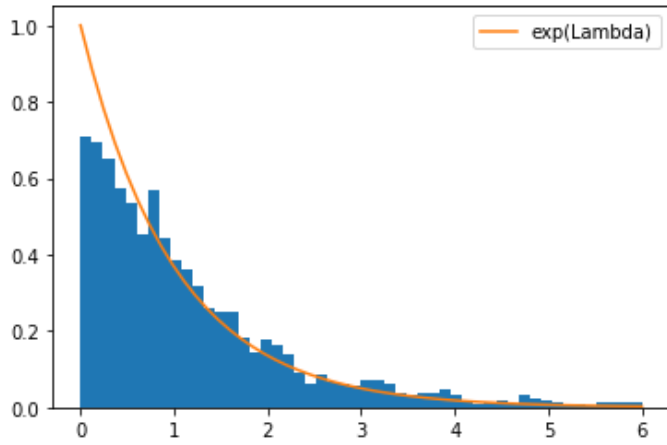


Количество потоков - n=2

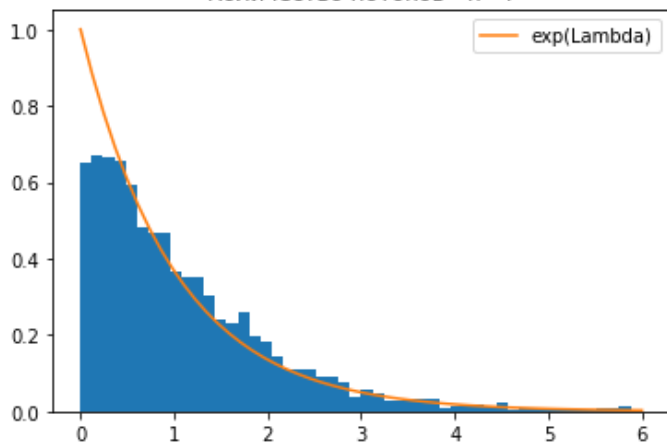




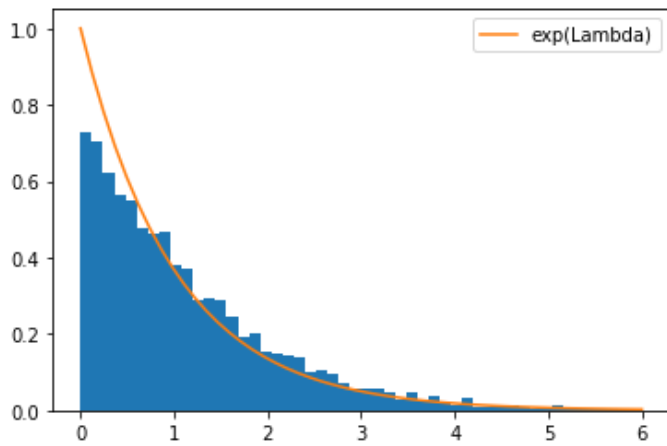
Количество потоков - $n=3$



Количество потоков - $n=4$

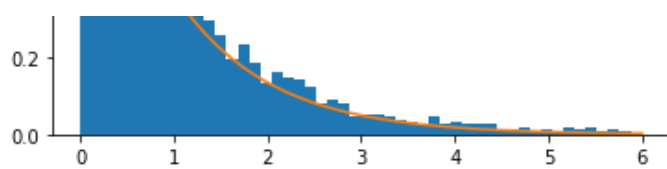


Количество потоков - $n=5$

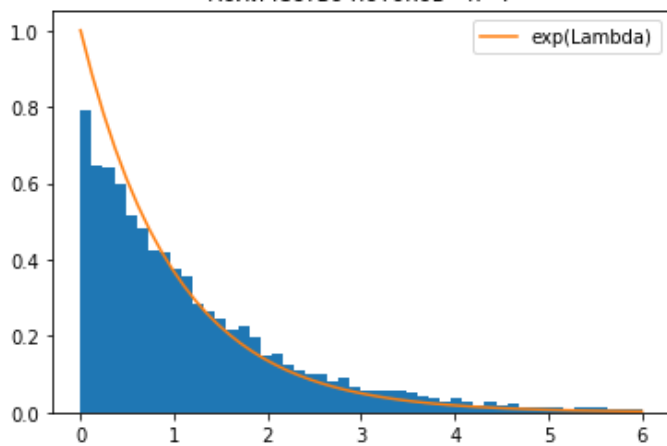


Количество потоков - $n=6$

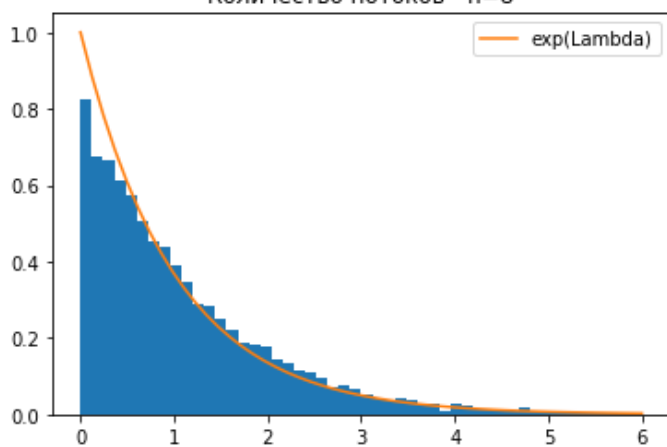




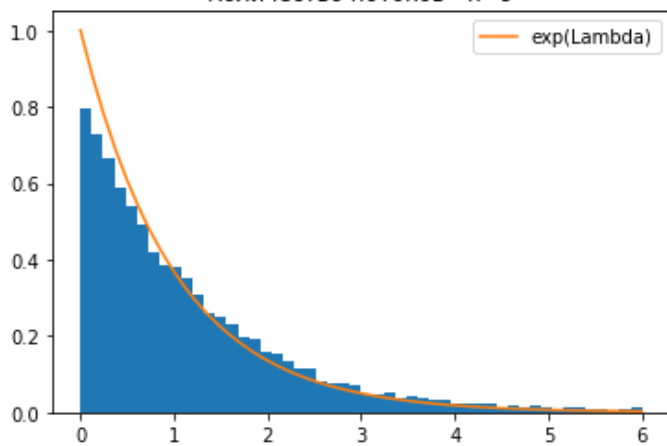
Количество потоков - $n=7$



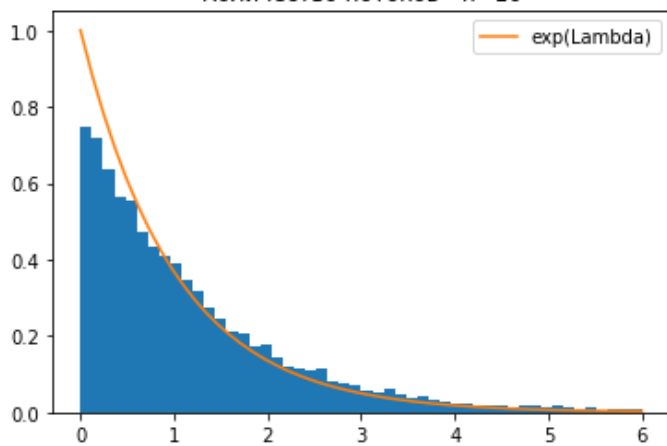
Количество потоков - $n=8$



Количество потоков - $n=9$



Количество потоков - $n=10$



Симуляция была проведена для двух распределений: для равномерного и для χ^2 . Гистограммы для обоих распределений становятся похожими на экспоненциальное с увеличением n .