

In [40]:

```
import numpy as np
import scipy.stats as sps
import matplotlib.pyplot as plt
import math
%matplotlib inline
```

In [41]:

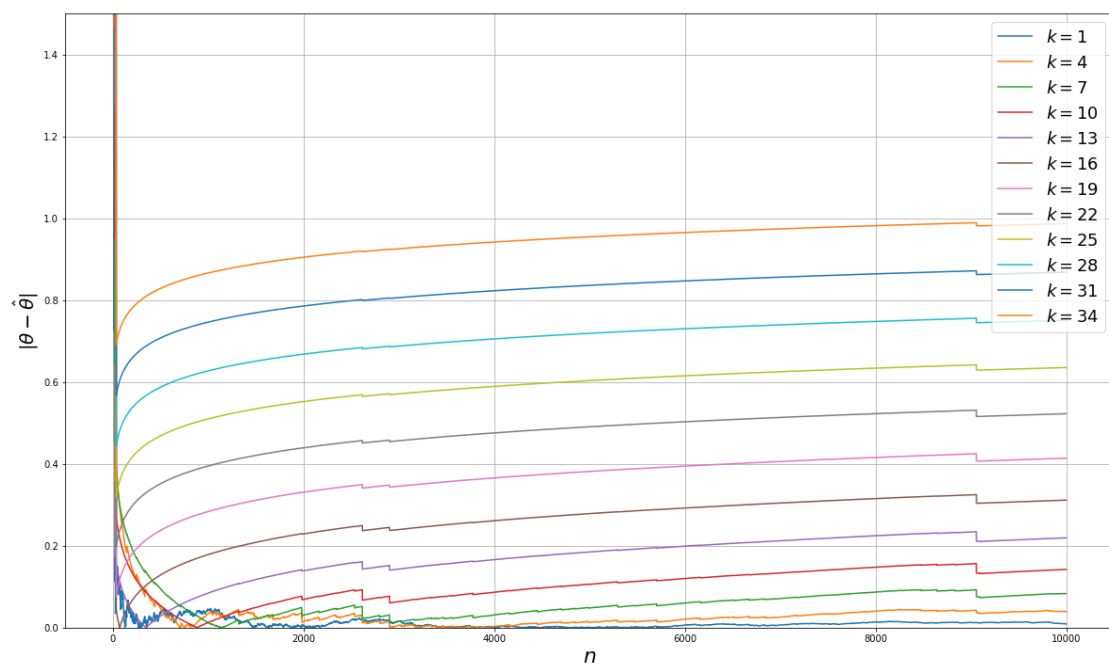
```
# вычисление оценки
def estimation(k):
    return [pow(math.factorial(k) / np.average(sample[:n] ** k), 1 /
k) for n in range(1, N)]
```

In [42]:

```
N = 10**4
theta = 1

# генерация выборки
sample = sps.expon.rvs(size = N, loc = 0, scale = theta)

# построение графиков для всех оценок
n = np.arange(1, N, dtype=int)
plt.figure(figsize = (20, 12))
for k in range (1, 35, 3):
    plt.plot(n, np.abs(estimation(k) - theta*np.ones(N-1)), label='$k$'+k.__str__())
plt.xlabel(r'$n$', fontsize = 22)
plt.ylabel(r'$| \theta - \hat{\theta} |$', fontsize = 20)
plt.legend(fontsize=18, loc=1)
plt.ylim(0, 1.5)
plt.grid()
plt.show()
```



Вывод:

По графику отчётливо видно, что наиболее хорошие оценки получаются при наименьших параметрах k .