Граница Грайсмера имеет вид: $N(k,d) \geq \sum_{i=0}^{k-1} ceil(\frac{d}{2^i})$

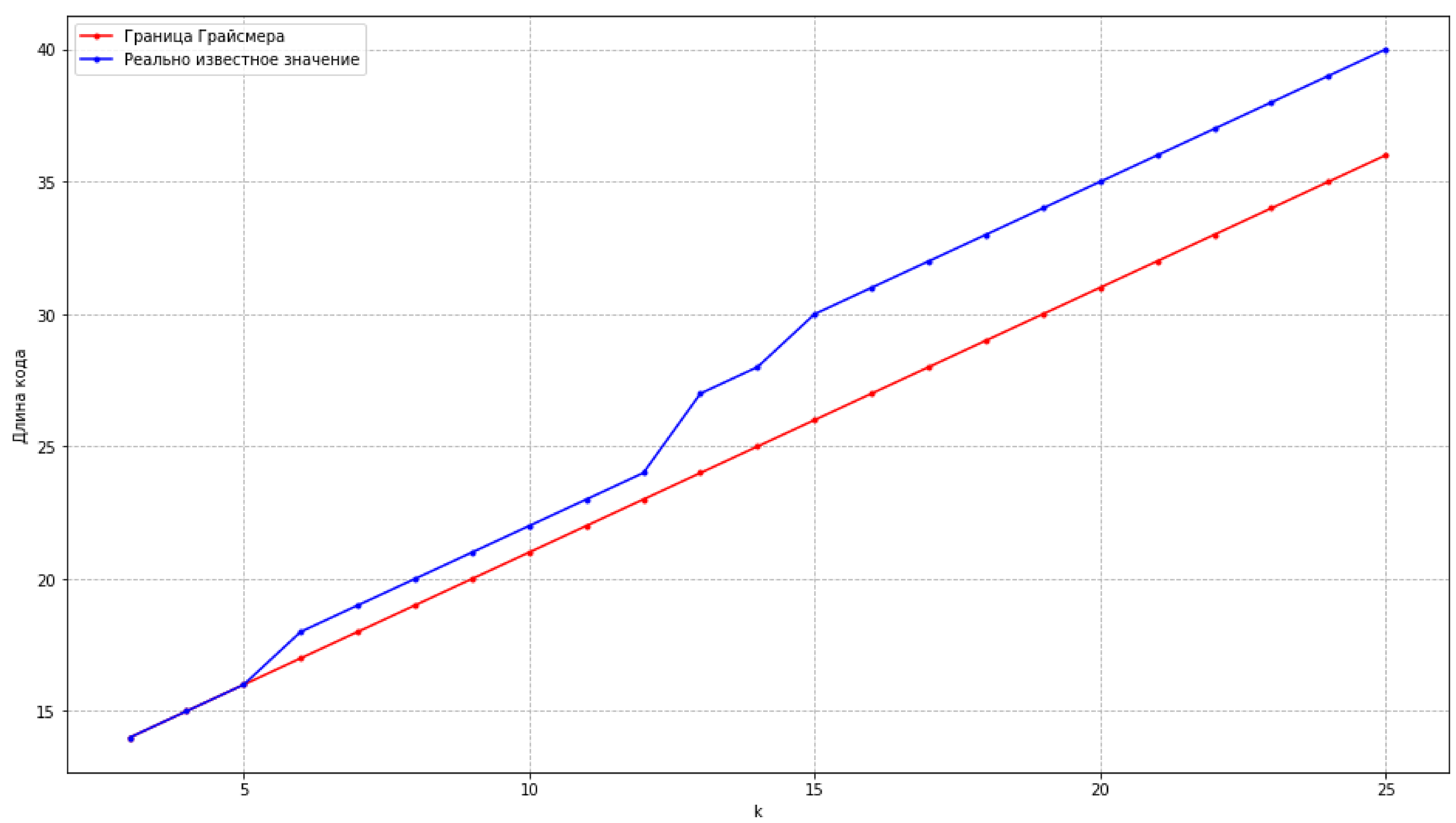
In [2]:
```python
import math
import matplotlib.pyplot as plt
```

In [12]:
```python
def get_graysmer_border(k, d):
    total_sum = 0
    for i in range(k):
        total_sum += math.ceil(d / (2 ** i))
    return total_sum
```

In [13]:
```python
d = 8
k_to_n = {
    3: 14,
    4: 15,
    5: 16,
    6: 18,
    7: 19,
    8: 20,
    9: 21,
    10: 22,
    11: 23,
    12: 24,
    13: 27,
    14: 28,
    15: 30,
    16: 31,
    17: 32,
    18: 33,
    19: 34,
    20: 35,
    21: 36,
    22: 37,
    23: 38,
    24: 39,
    25: 40
}
ks = list(k_to_n.keys())
ks.sort()
ns = []
graysmers = []
```

In [14]:
```python
for k in ks:
    n = k_to_n[k]
    graysmer = get_grajsmer_border(k, 8)
    ns.append(n)
    graysmers.append(graysmer)
```

In [15]:
```python
plt.figure(figsize=(16,9))
plt.grid(linestyle='--')
plt.plot(ks, graysmers, linestyle='-',marker='.',color='r', label='Граница Грайсмера')
plt.plot(ks, ns, linestyle='-',marker='.',color='b', label='Реально известное значение')
plt.xlabel('k')
plt.ylabel('Длина кода')
plt.legend()
plt.show()
```



Как видно, при маленьких k граница Грайсмра достаточно хорошо описывает минимально возможную длину кода, однако с ростом k реальная известная длина начинает отдоляться от границы. Это можно объяснить тем, что знаменатель 2^i довльно быстро растёт в силу своей экспоненциальности. Из-за этого он довльно быстро начинает привосходить d и с увеличением k значение границы начинает расти лишь на 1, когда в реальности это оказывется не так.

Например, при $k = 28, d = 8$ в реальности расстояние хотя бы 45, когда как граница Грайсмера даёт оценку $n \geq 39$. Однако же точно извествое минимальное расстояние для $(39, 28)$-кода равно 5.

In [ ]: