In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [ ]:
```
algorithms = [
    "naive",
    "kmp-standard",
    "kmp-refined",
]

texts = [
    "standard-small-bin",
    "standard-small-sq",
    "standard-big-bin",
    "standard-big-sq",
]
```

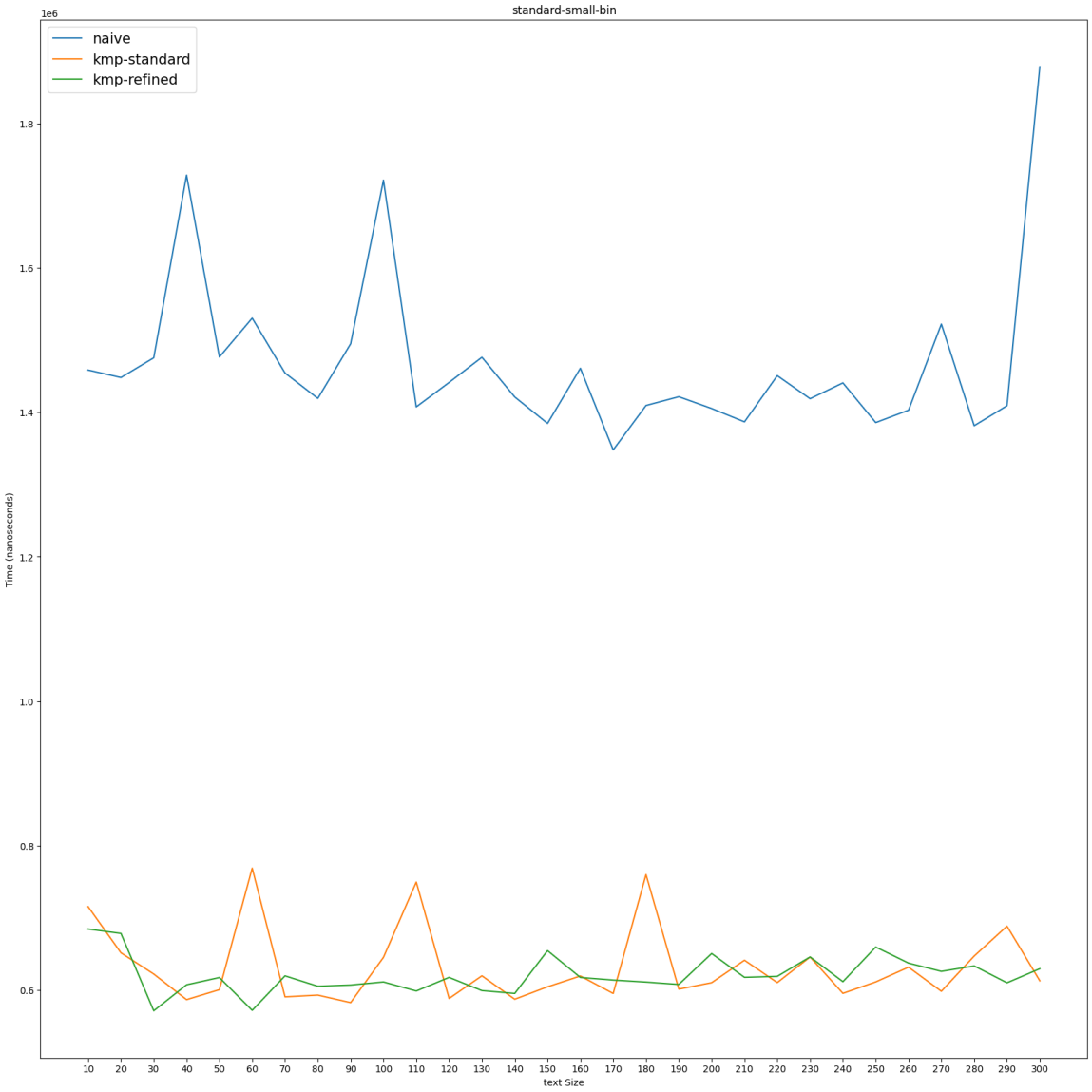
In [ ]:
```
data = pd.read_csv('../data.csv', sep=';', header=None)
data.columns = ['algorithm', 'text', 'size', 'time']
```

In [ ]:
```
def print_text(data, text):
    text_df = data[data['text'] == text]
    plt.figure(figsize=(20, 20))
    for algorithm in algorithms:
        df = text_df[text_df['algorithm'] == algorithm]
        plt.plot(df['size'], df['time'], label=algorithm)
    plt.title(text)
    plt.xlabel('text Size')
    plt.xticks(text_df['size'].unique())
    plt.ylabel('Time (nanoseconds)')
    plt.legend(labelcolor='black', prop={'size': 15})
```

Зависимость времени выполнения от размера паттерна

Без символов подстановки, 10000 символов, бинарный алфавит

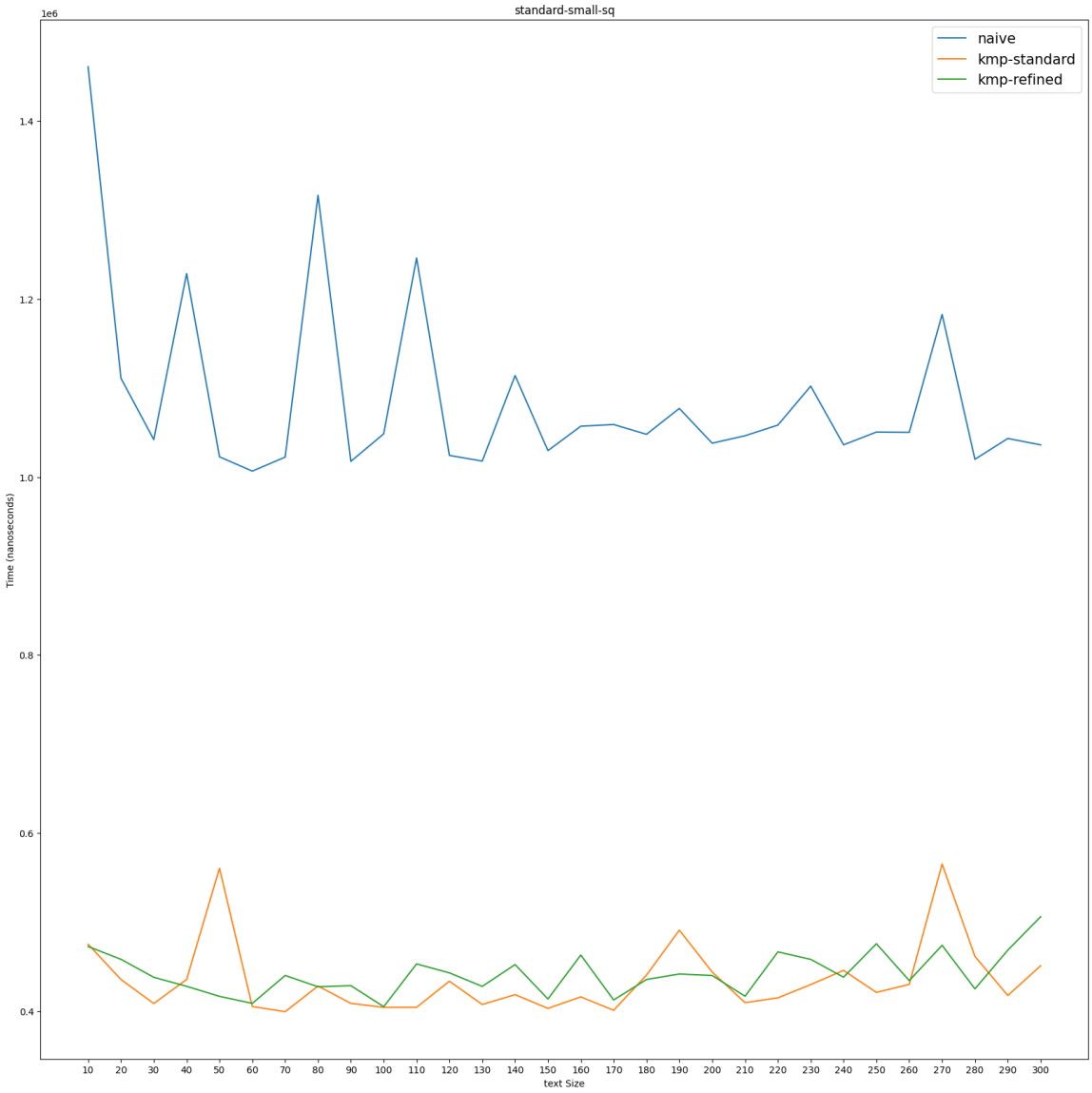
In [ ]:
```
print_text(data, texts[0])
```



Вывод: TBA

2. Без символов подстановки, 10000 символов, 4-символьный алфавит

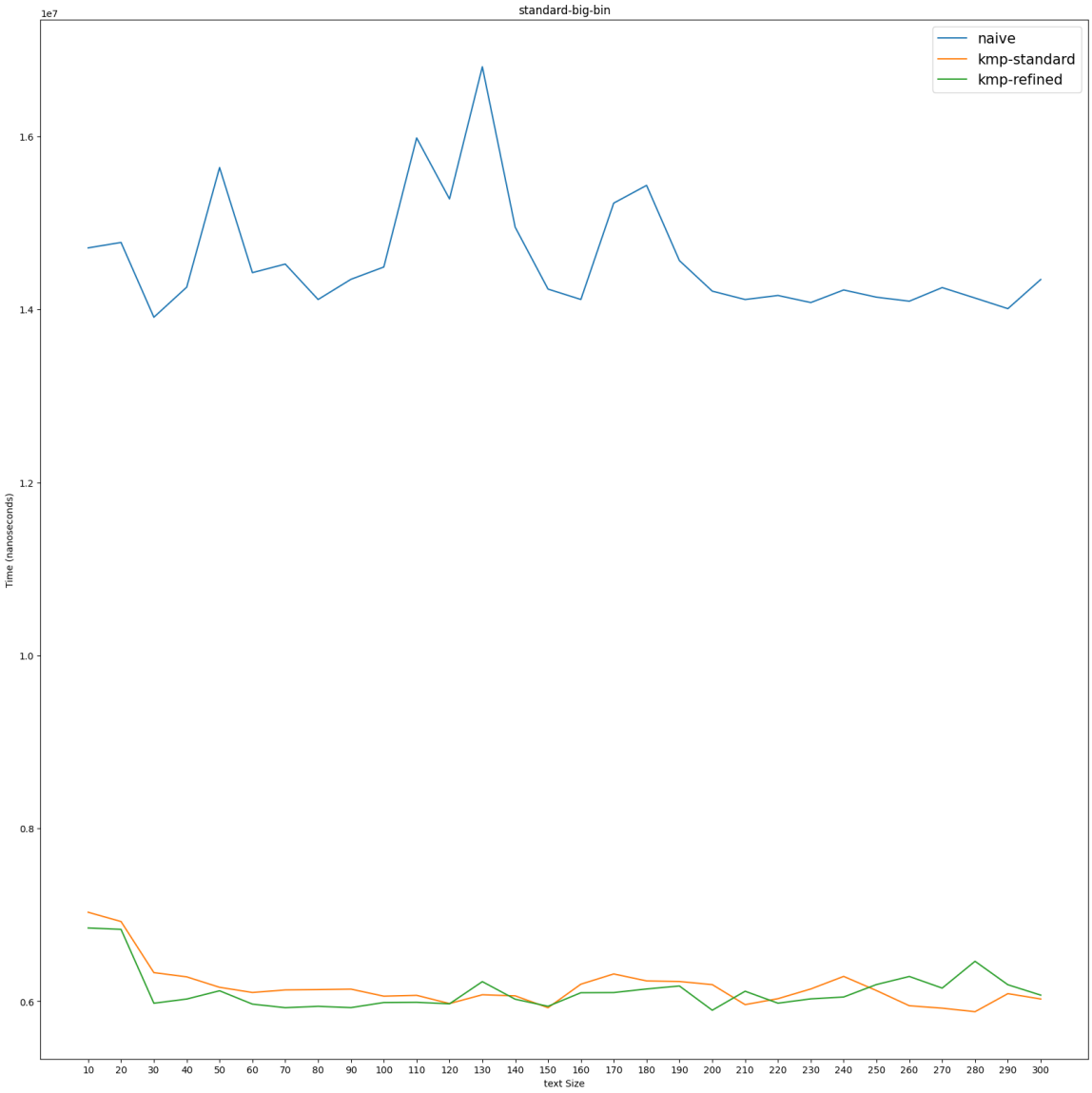
In [ ]:
```
print_text(data, texts[1])
```



Вывод: TBA

3. Без символов подстановки, 100000 символов, бинарный алфавит

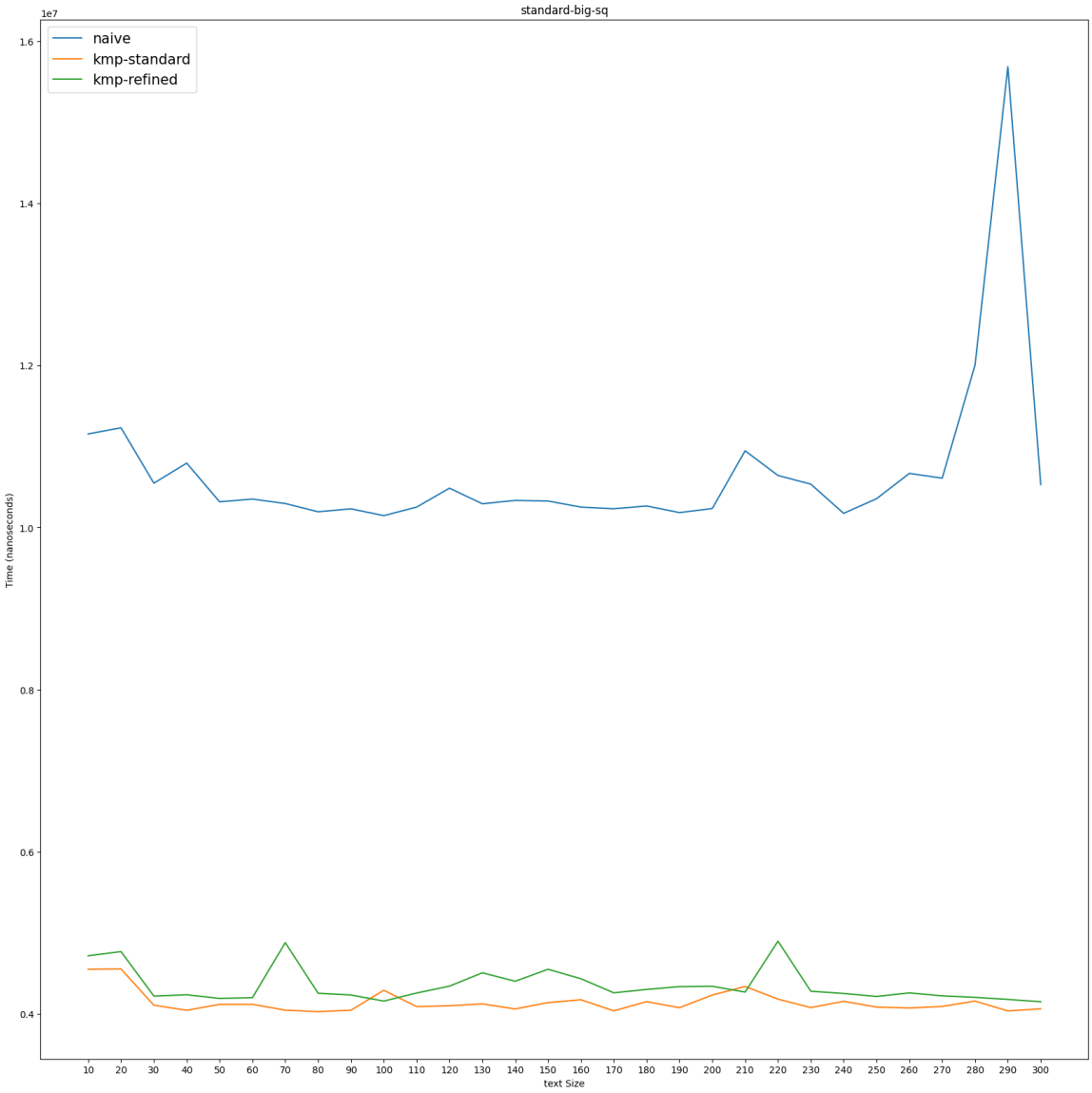
In [ ]:
```
print_text(data, texts[2])
```



Вывод: TBA

4. Без символов подстановки, 100000 символов, 4-символьный алфавит

In [ ]:
```
print_text(data, texts[3])
```


standard-big-sq
1e7
naive
kmp-standard
kmp-refined
Time (nanoseconds)
text Size

Вывод: TBA