

# analysis

May 8, 2024

## 1 A1

1.0.1 /data/tests/

1.0.2 /data/

1.0.3 :

- 

1.0.4 StringGenerator - ,

—

- 

1.0.5 :

—

—

—

—

—

—

—

—

```
[36]: from itertools import chain
import matplotlib.pyplot as plt
```

```
[28]: class SortLog:
    def __init__(self, time, comparisons, is_sorted):
        self.time = time
        self.comparisons = comparisons
        self.is_sorted = is_sorted

    def check_all_sorted(logs: list[SortLog]) -> bool:
        return all(i.is_sorted for i in logs)

    def parse(s: str) -> SortLog:
        log = SortLog()
        s = s[1:-2]
        s = s.split(',')
        log.time = int(s[0].split()[1][:-2])
        log.comparisons = int(s[1].strip().split()[1])
        log.is_sorted = (s[2].strip().split()[1] == 'true')
        return log

    def get_logs(path: str) -> list[SortLog]:
        logs = []
        with open(path, 'r') as file:
            for i in file.readlines():
                logs.append(parse(i))
        return logs
```

...

```
[26]: q_sort = [
    get_logs("../data/q_sort_random.txt"),
    get_logs("../data/q_sort_backward.txt"),
    get_logs("../data/q_sort_almost_sorted.txt")
]

merge_sort = [
    get_logs("../data/merge_sort_random.txt"),
    get_logs("../data/merge_sort_backward.txt"),
    get_logs("../data/merge_sort_almost_sorted.txt")
]

q_sort_string = [
    get_logs("../data/q_sort_string_random.txt"),
    get_logs("../data/q_sort_string_backward.txt"),
    get_logs("../data/q_sort_string_almost_sorted.txt")
]

merge_sort_string = [
    get_logs("../data/merge_sort_string_random.txt"),
```

```

    get_logs("../data/merge_sort_string_backward.txt"),
    get_logs("../data/merge_sort_string_almost_sorted.txt")
]
msd_basic = [
    get_logs("../data/msd_radix_sort_random.txt"),
    get_logs("../data/msd_radix_sort_backward.txt"),
    get_logs("../data/msd_radix_sort_almost_sorted.txt"),
]
msd_switched = [
    get_logs("../data/msd_radix_sort_switched_random.txt"),
    get_logs("../data/msd_radix_sort_switched_backward.txt"),
    get_logs("../data/msd_radix_sort_switched_almost_sorted.txt"),
]

```

:

```

[32]: check_all_sorted(list(
    chain.from_iterable(
        chain.from_iterable([q_sort, merge_sort, q_sort_string,
↪merge_sort_string, msd_basic, msd_switched])))

```

[32]: True

```

[58]: def get_comparisons(i: SortLog):
    return i.comparisons

def get_time(i: SortLog):
    return i.time

def plott(axis, y, what, color, label):
    axis.scatter(list(i for i in range(100, 3000 + 1, 100)), list(what(i) for i
↪in y), color=color, label=label)

def name_of(what):
    if what == get_time:
        return "          "
    else:
        return "          "

def plot(array, name, what):
    fig, ax = plt.subplots()
    plott(ax, array[0], what, "red", f"{name} random")

```

```

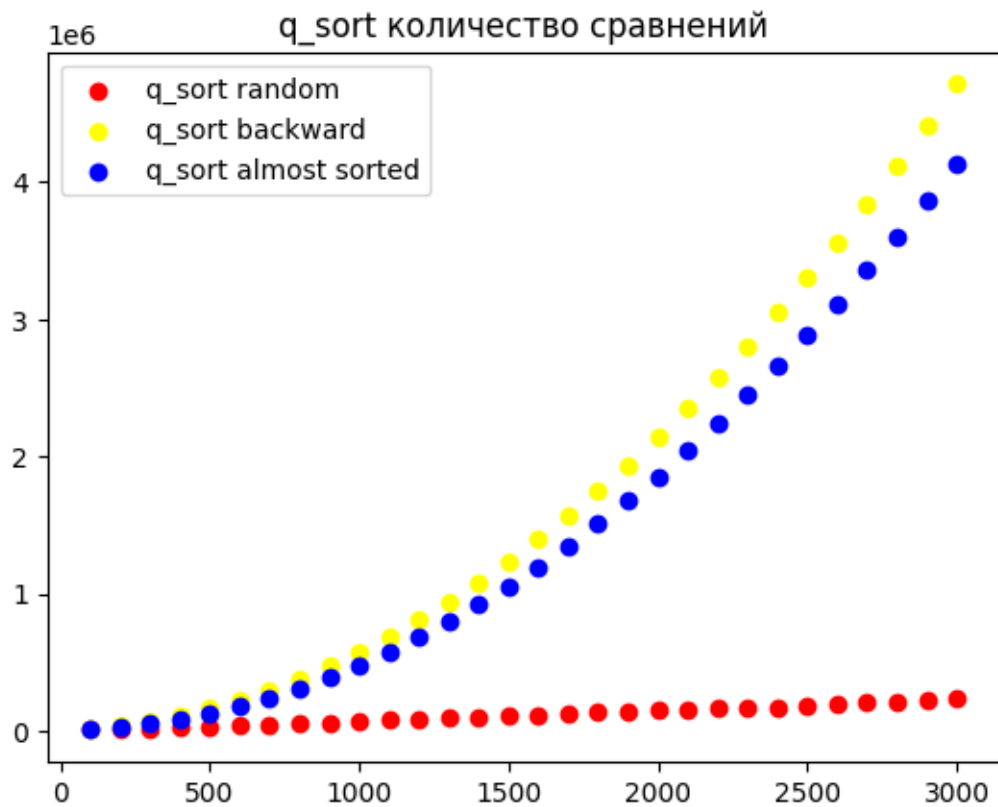
plott(ax, array[1], what, "yellow", f"{name} backward")
plott(ax, array[2], what, "blue", f"{name} almost sorted")
ax.legend()
plt.title(f"{name} {name_of(what)}")
plt.show()

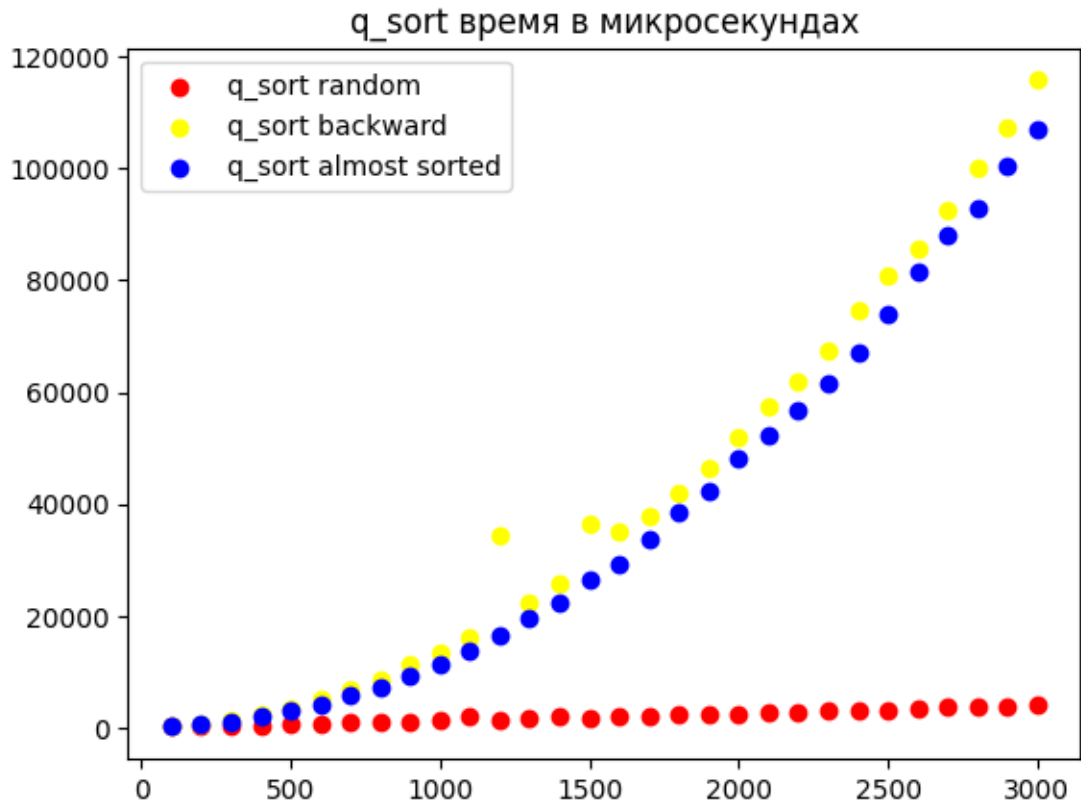
```

```

[60]: plot(q_sort, "q_sort", get_comparisons)
      plot(q_sort, "q_sort", get_time)

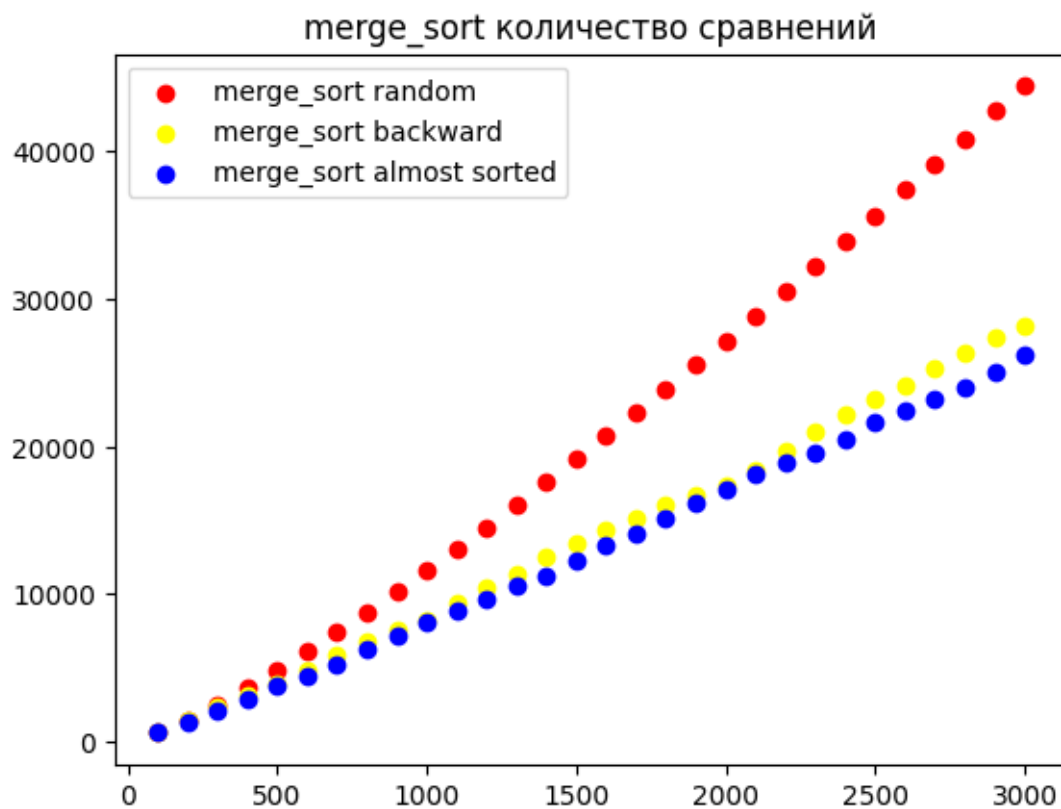
```

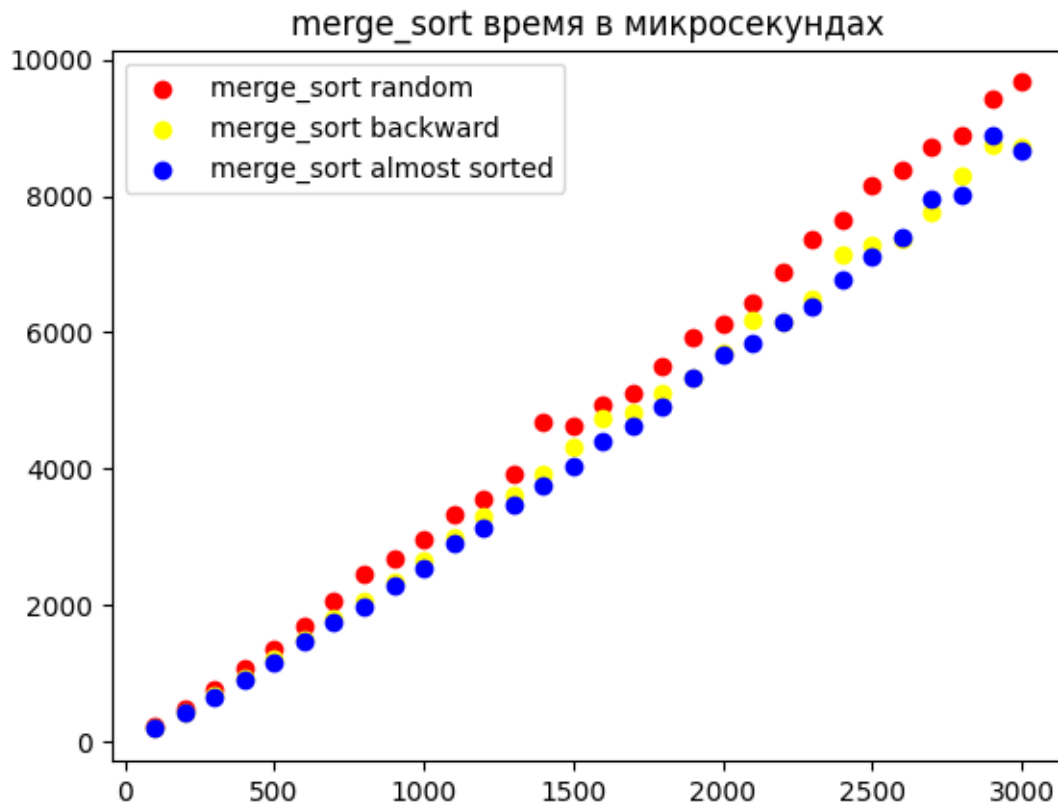




, ( ). q\_sort - , pivot  
 $O(n^2)$

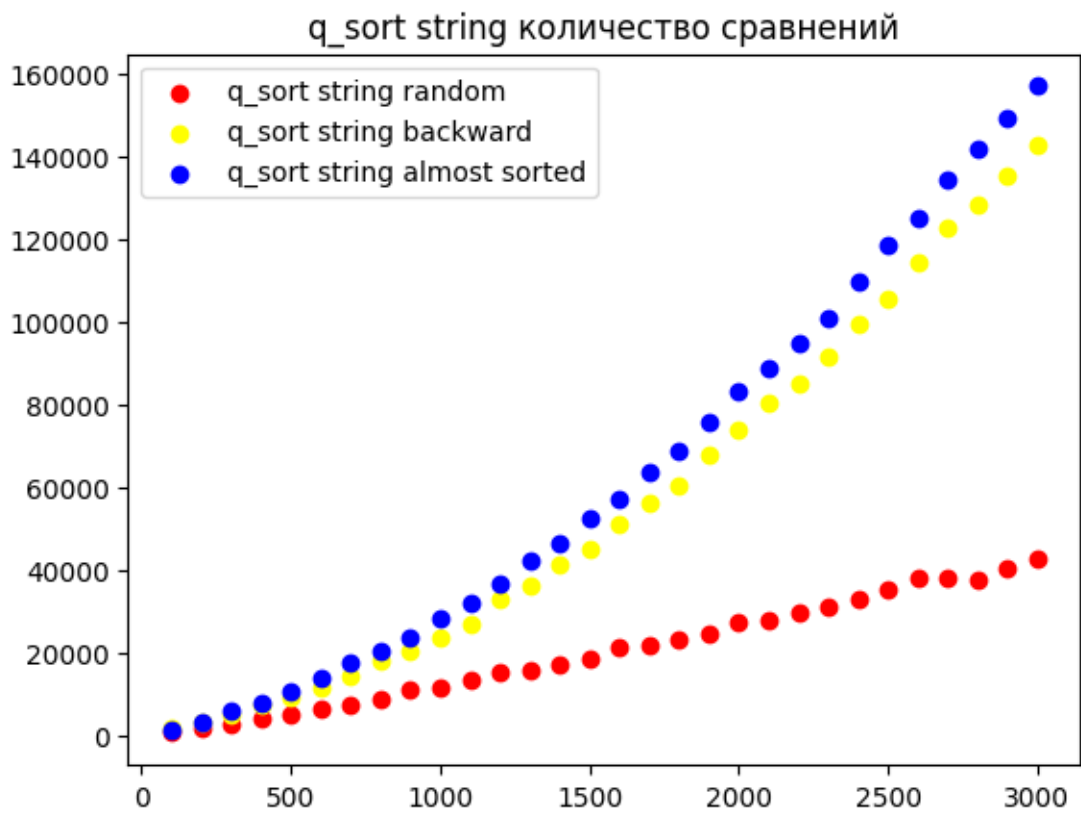
```
[61]: plot(merge_sort, "merge_sort", get_comparisons)
      plot(merge_sort, "merge_sort", get_time)
```



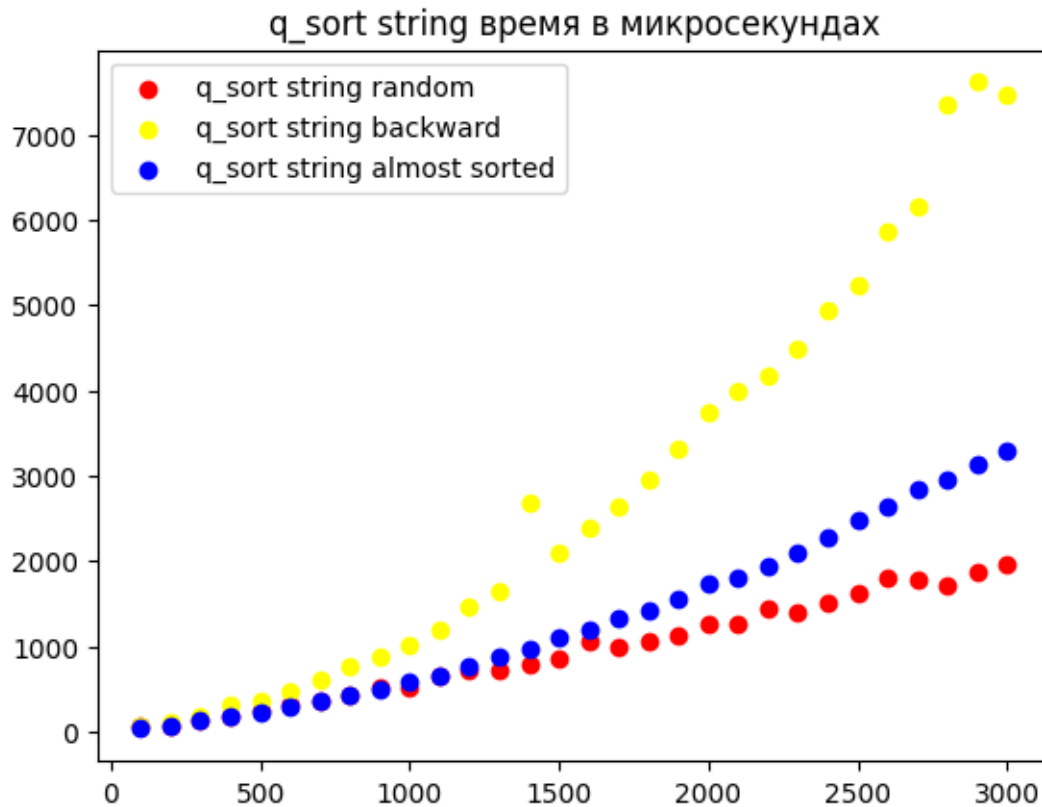


quick sort. / / - merge sort quick sort. -  
 , merge sort  
 - merge sort “ ”.

```
[62]: plot(q_sort_string, "q_sort string", get_comparisons)
      plot(q_sort_string, "q_sort string", get_time)
```

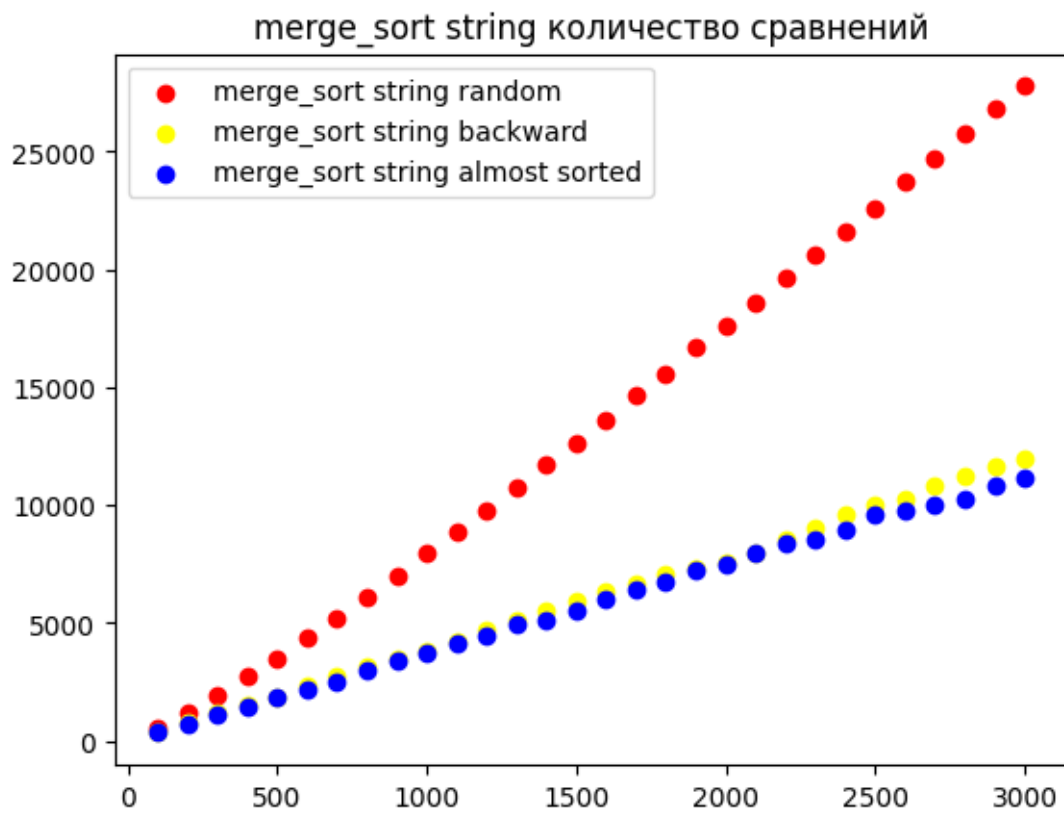


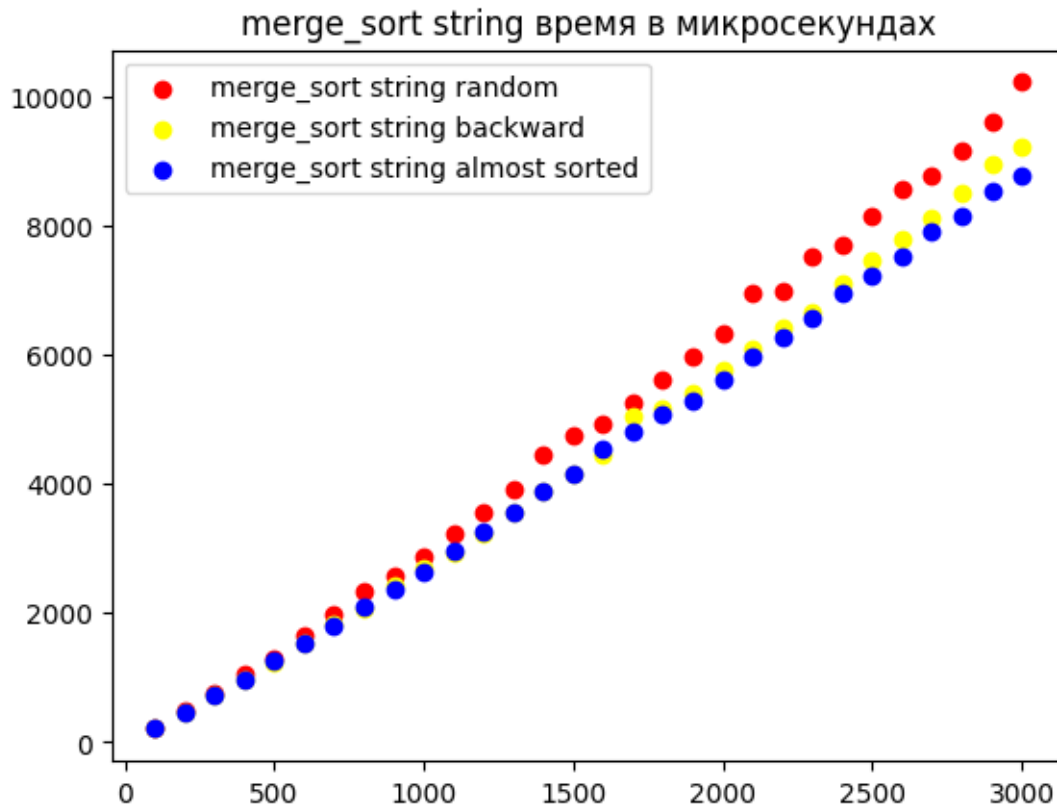




```
qsort 4'714'009 ( backwards, 3000 ). quick sort
160'000.
,
pivot',
```

```
[63]: plot(merge_sort_string, "merge_sort string", get_comparisons)
plot(merge_sort_string, "merge_sort string", get_time)
```



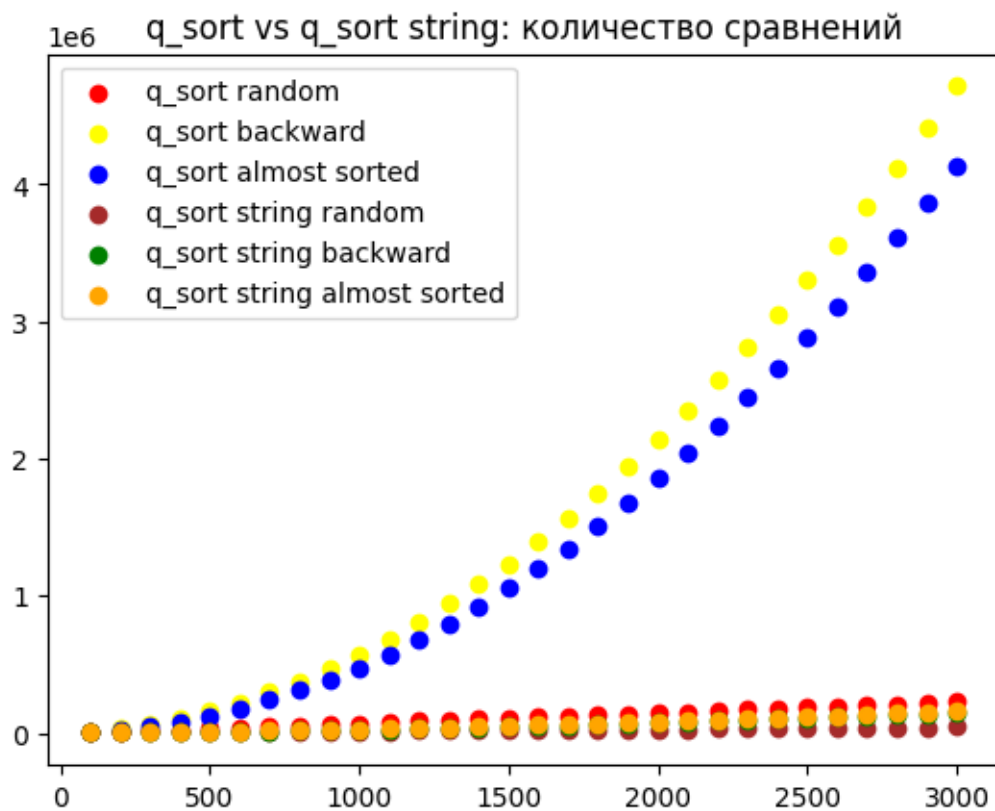


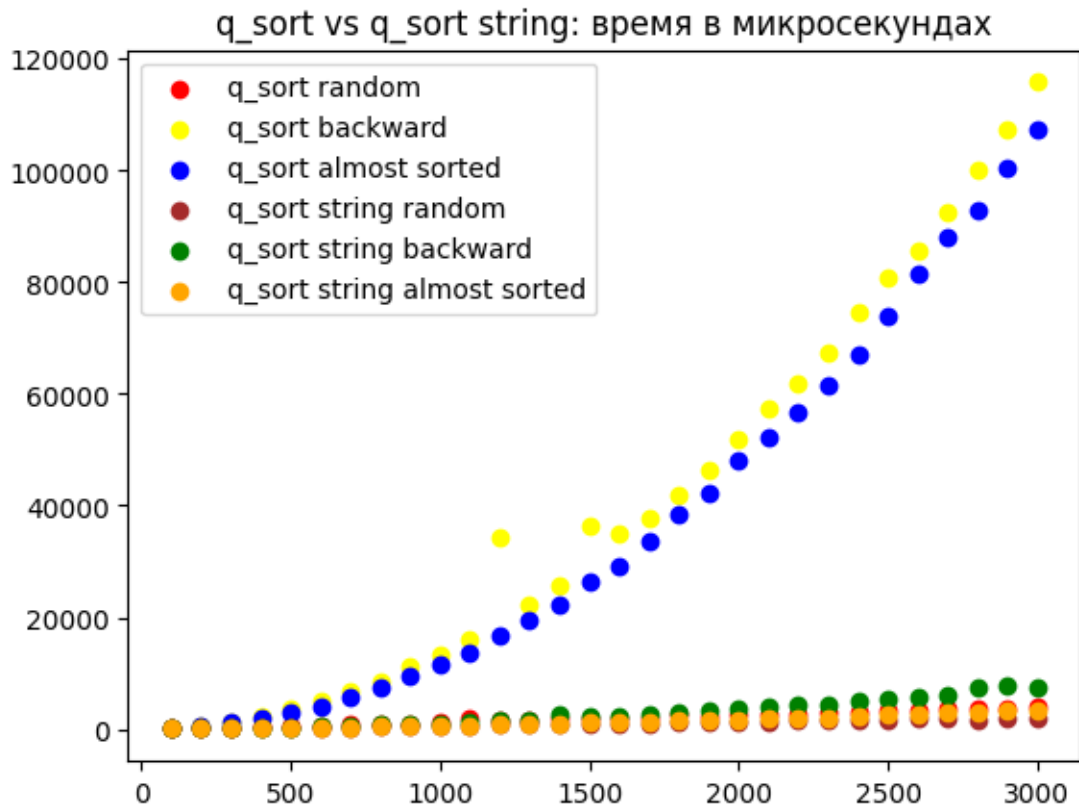
merge\_sort string , - lcp

qsort merge\_sort :

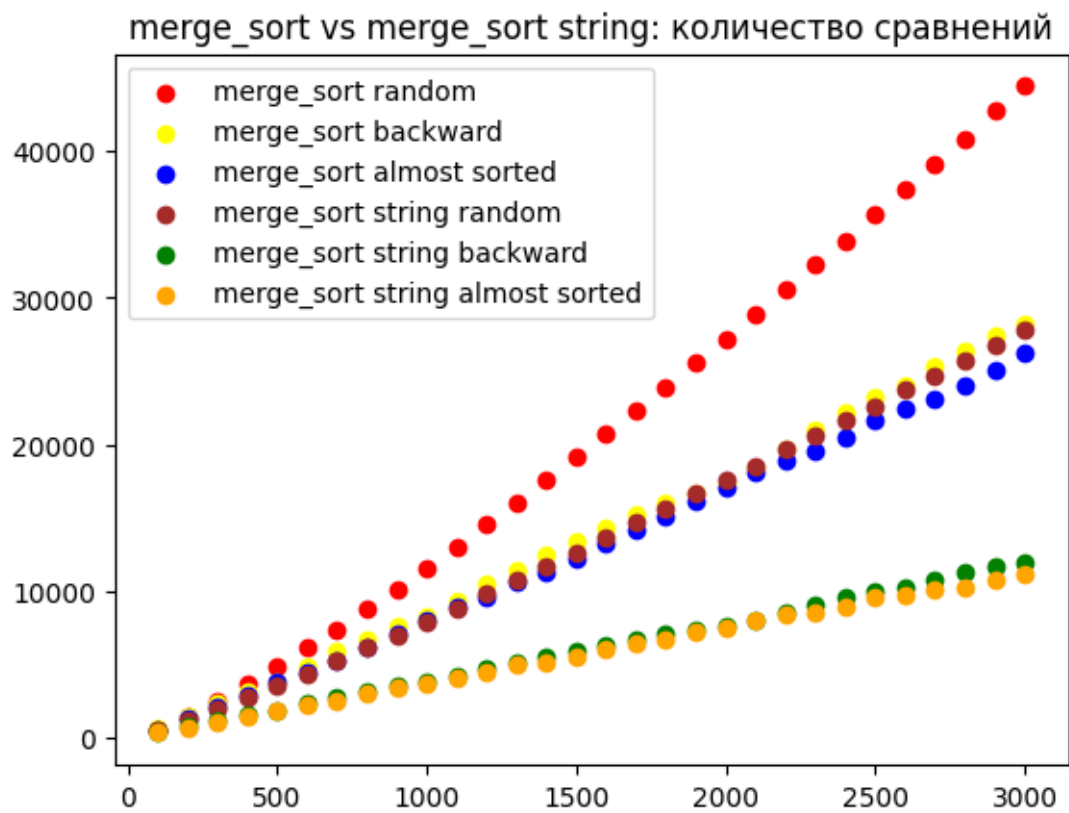
```
[66]: def compare(arr1, name1, arr2, name2, what):
    fig, ax = plt.subplots()
    plott(ax, arr1[0], what, "red", f"{name1} random")
    plott(ax, arr1[1], what, "yellow", f"{name1} backward")
    plott(ax, arr1[2], what, "blue", f"{name1} almost sorted")
    plott(ax, arr2[0], what, "brown", f"{name2} random")
    plott(ax, arr2[1], what, "green", f"{name2} backward")
    plott(ax, arr2[2], what, "orange", f"{name2} almost sorted")
    ax.legend()
    plt.title(f"{name1} vs {name2}: {name_of(what)}")
    plt.show()

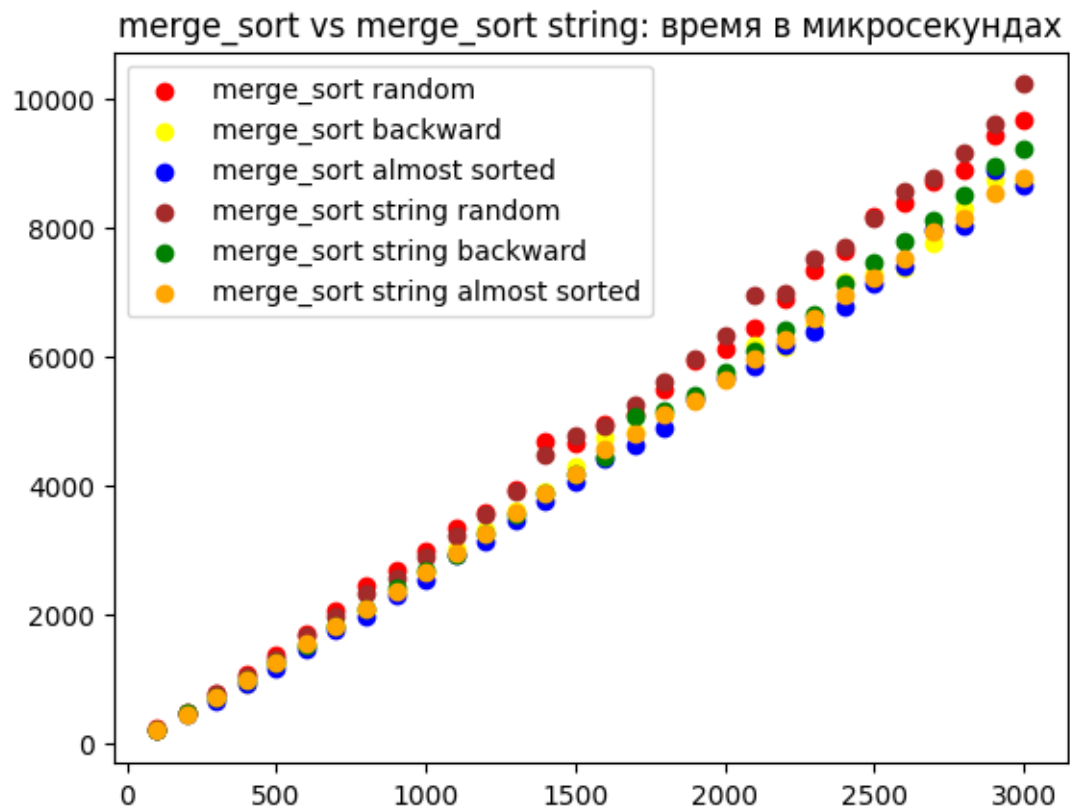
[68]: compare(q_sort, "q_sort", q_sort_string, "q_sort string", get_comparisons)
       compare(q_sort, "q_sort", q_sort_string, "q_sort string", get_time)
```



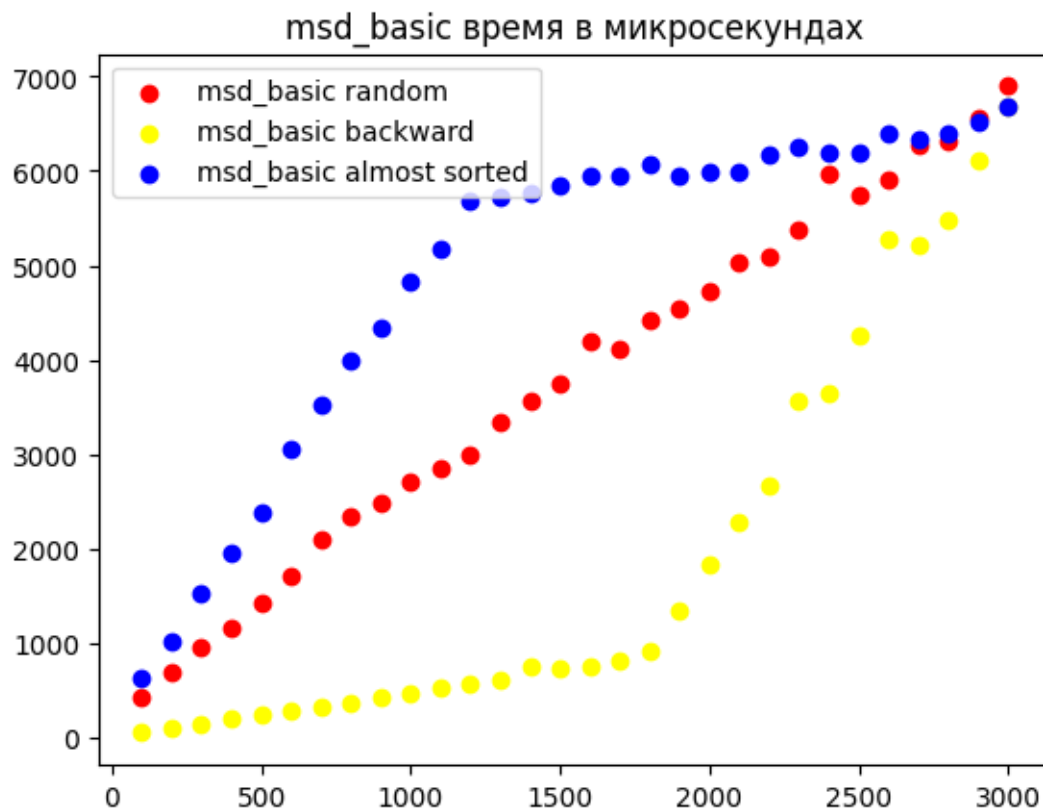


```
[69]: compare(merge_sort, "merge_sort", merge_sort_string, "merge_sort string",
    ↪get_comparisons)
compare(merge_sort, "merge_sort", merge_sort_string, "merge_sort string",
    ↪get_time)
```



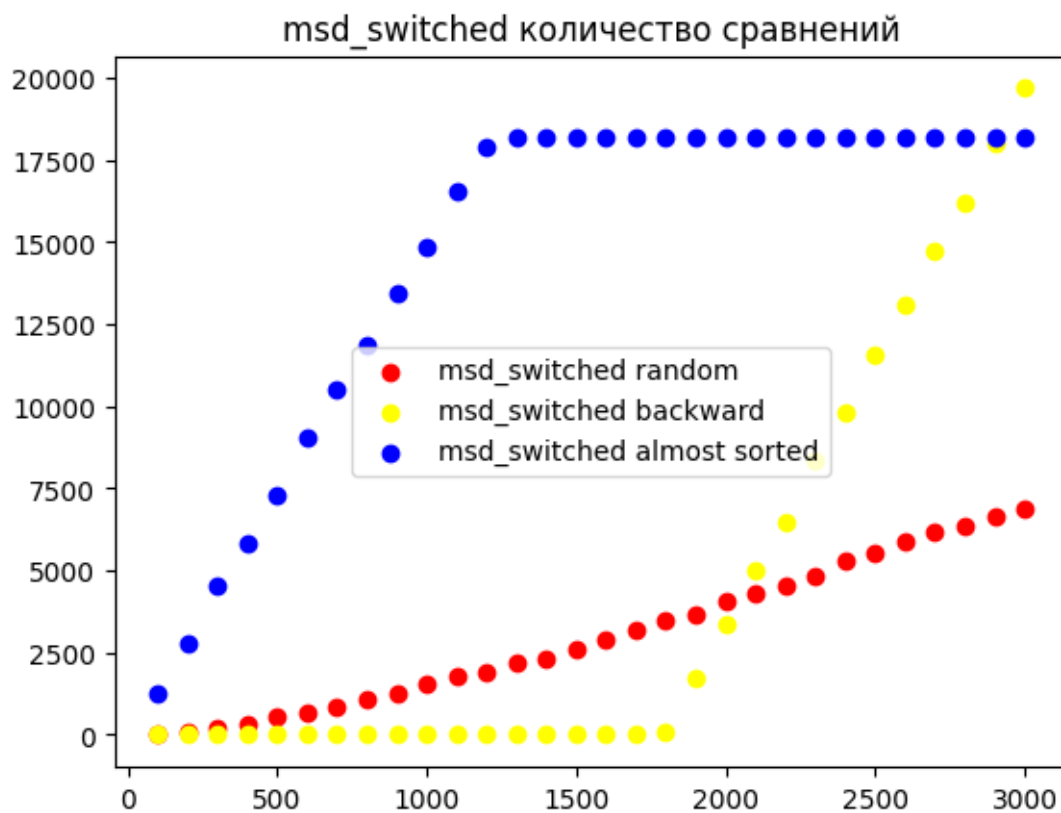


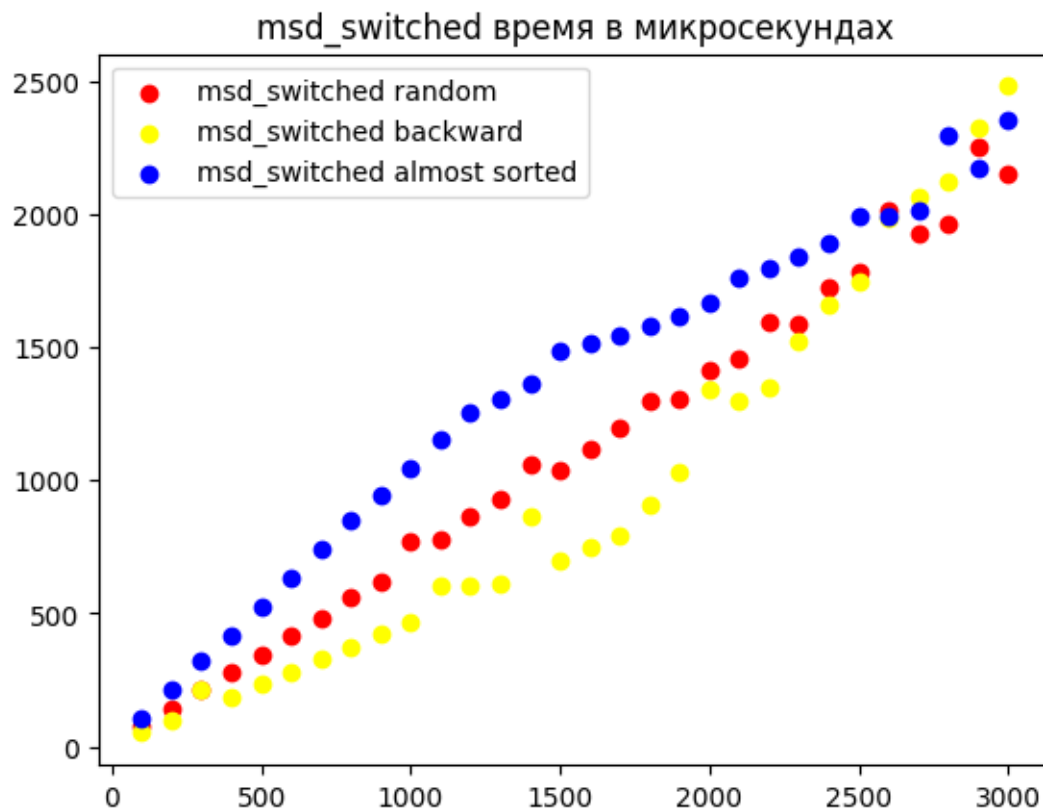
```
[71]: #plot(msd_basic, "msd_basic", get_comparisons) -  
      plot(msd_basic, "msd_basic", get_time)
```



```
[72]: plot(msd_switched, "msd_switched", get_comparisons)
      plot(msd_switched, "msd_switched", get_time)
```







```
[74]: #compare(msd_basic, "msd_basic", msd_switched, "msd_switched", get_comparisons)
      compare(msd_basic, "msd_basic", msd_switched, "msd_switched", get_time)
```



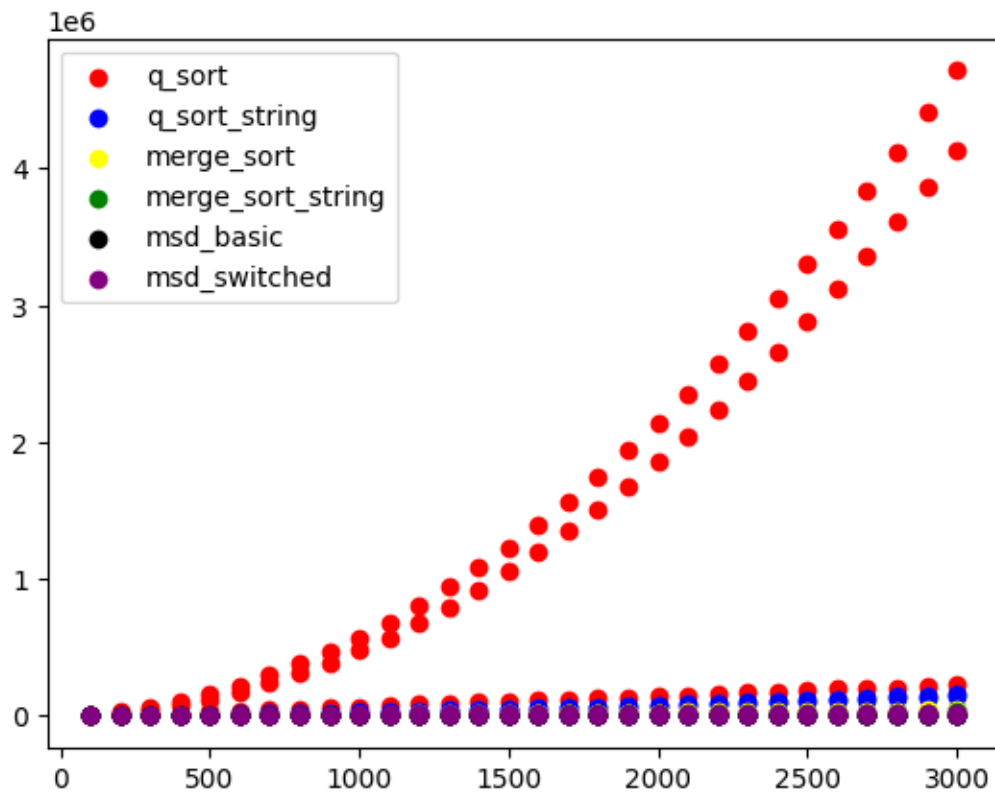
```

msd_switched
]
colors = [
    "red",
    "blue",
    "yellow",
    "green",
    "black",
    "purple"
]
labels = [
    "q_sort",
    "q_sort_string",
    "merge_sort",
    "merge_sort_string",
    "msd_basic",
    "msd_switched"
]

```

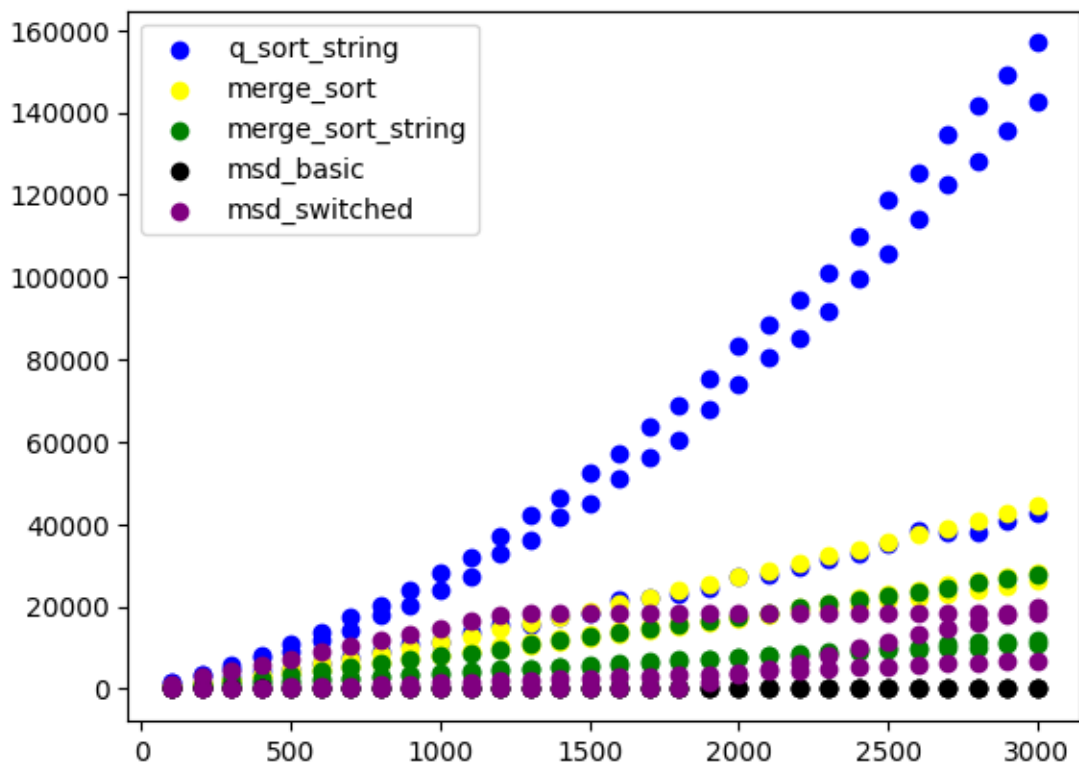
:

```
[150]: plot_all(data, colors, labels, get_comparisons)
```



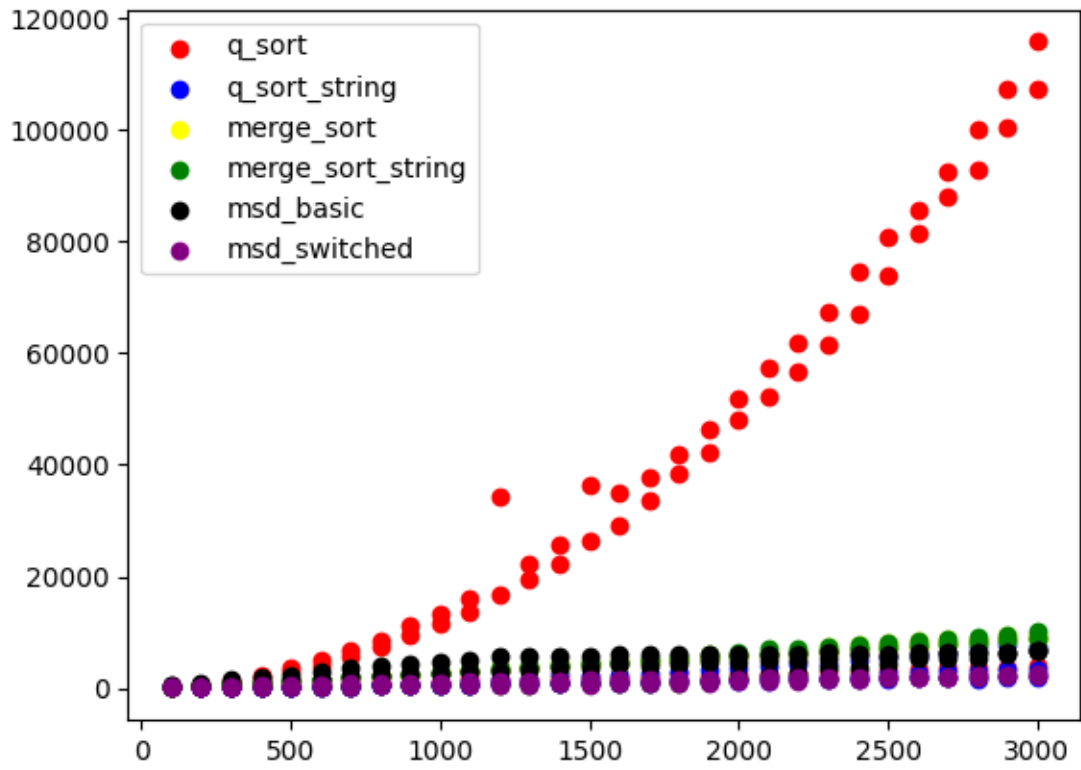
```
, - qsort'a, :
```

```
[151]: data1 = data.copy()
data1.pop(0)
colors1 = colors.copy()
colors1.pop(0)
labels1 = labels.copy()
labels1.pop(0)
plot_all(data1, colors1, labels1, get_comparisons)
```



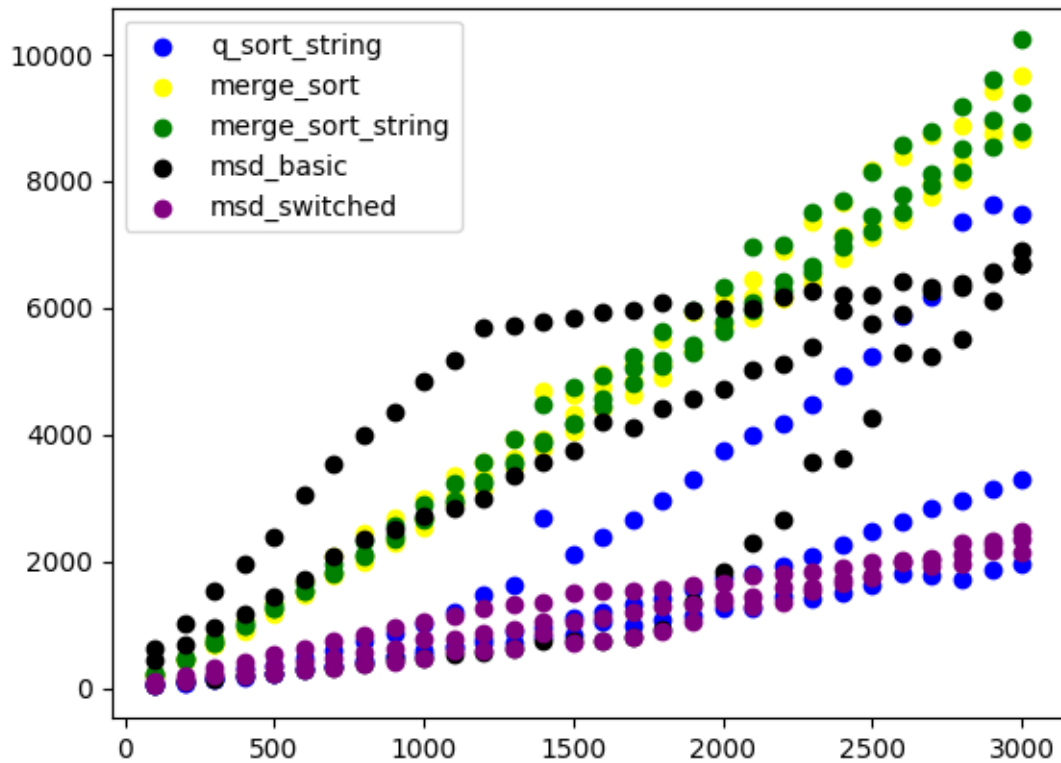
```
:
```

```
[152]: plot_all(data, colors, labels, get_time)
```



, qsort:

```
[153]: plot_all(data1, colors1, labels1, get_time)
```



```

(
    -
) -
qsort.
    -
    , merge_sort
    , merge_sort.
    , merge_sort
    , qsort.
    :
    q_sort -
    .
qsort.
    merge sort'
    ( -
    ), qsort
string
    -
    pivot'.
    .
    merge_sort -
    qsort string -
    (
    =>
    )
    ,
    .

```