range_stamps

June 19, 2024

```
[2]: import numpy as np
     import pandas as pd
     %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
[3]: arr = np.loadtxt('stamps.dat')
```

```
[8]: rarr = arr.reshape(int(len(arr)/5),5)
```

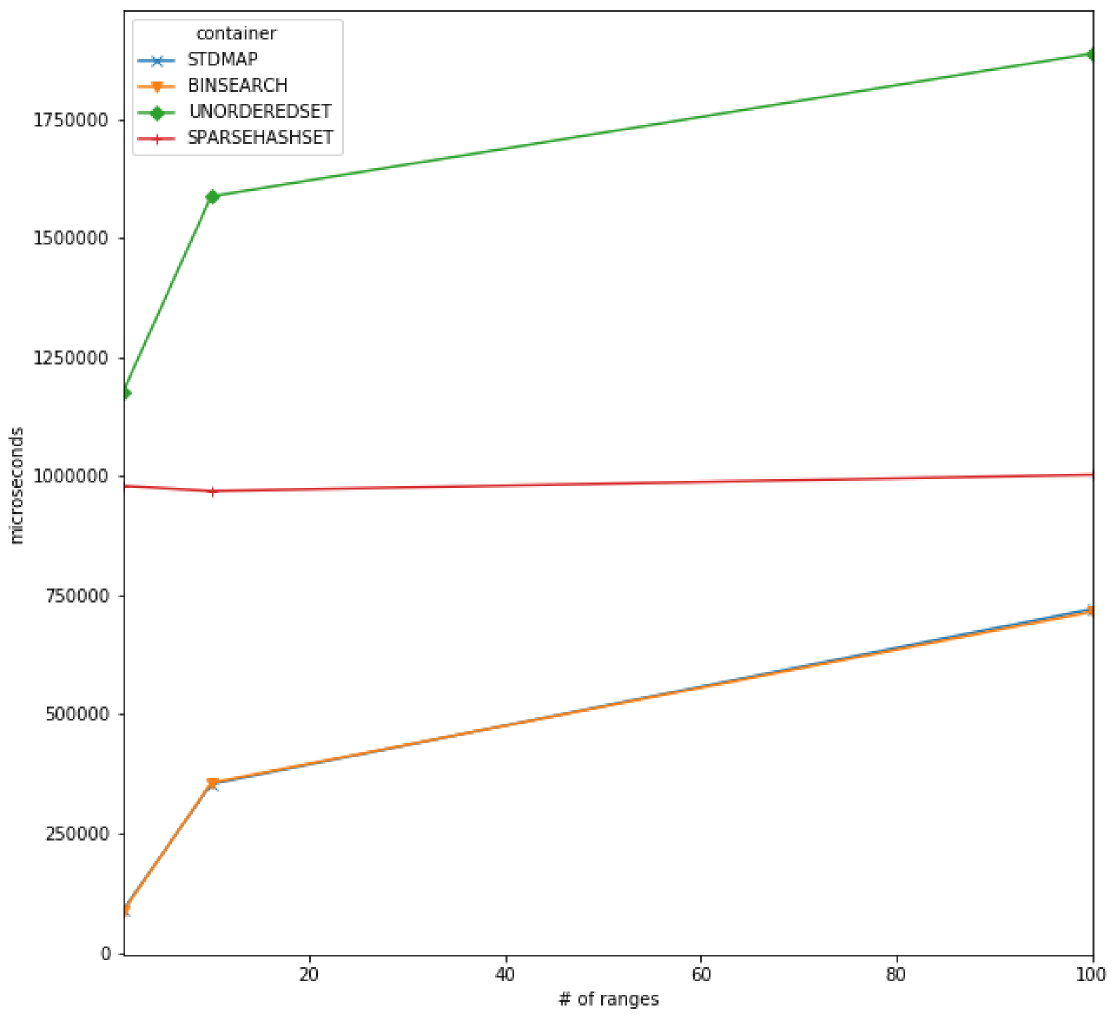
```
[9]: frame = pd.DataFrame(rarr)
```

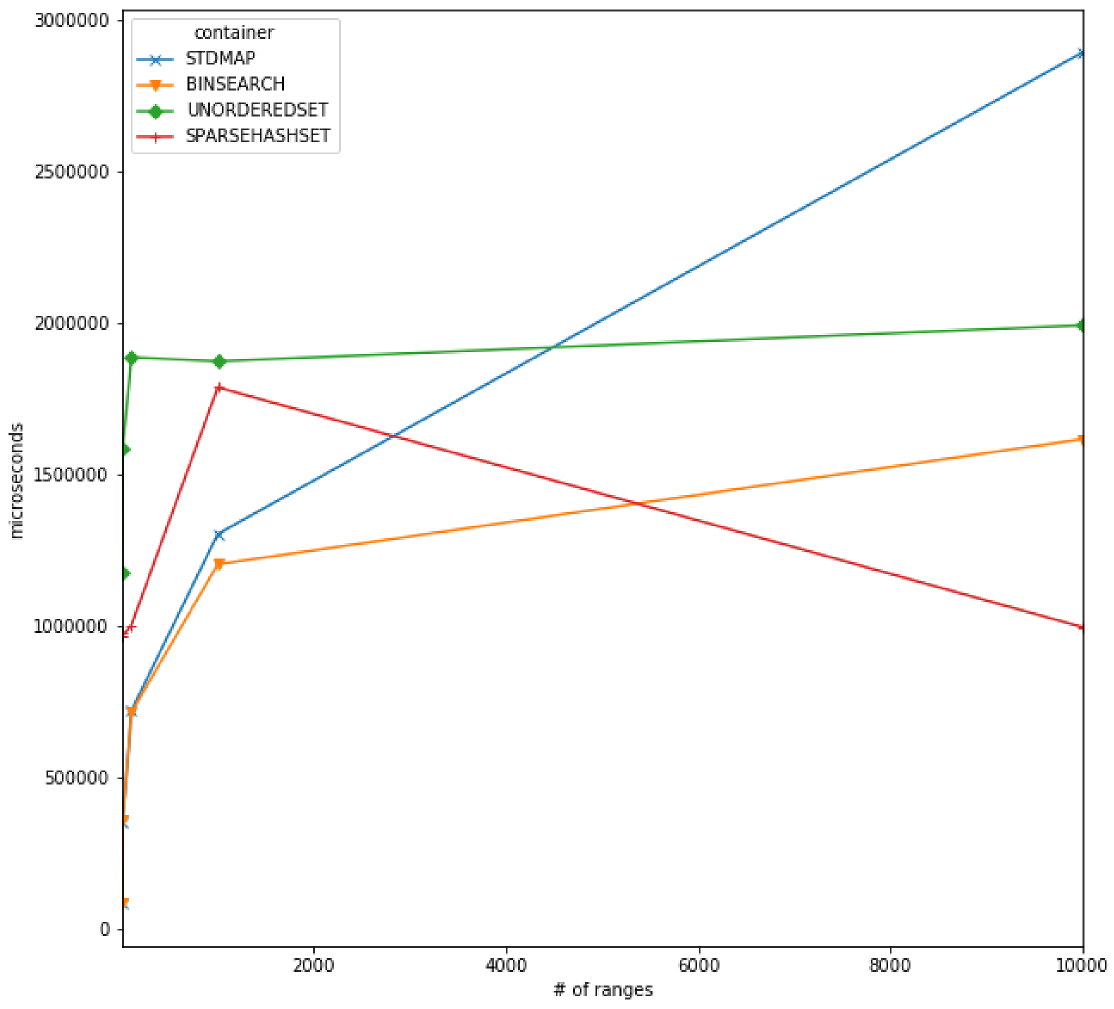
```
[10]: frame.columns = ['ranges','STDMAP', 'BINSEARCH', 'UNORDEREDSET',␣
       ↪'SPARSEHASHSET']
      frame.set_index('ranges', inplace=True)
      frame
```

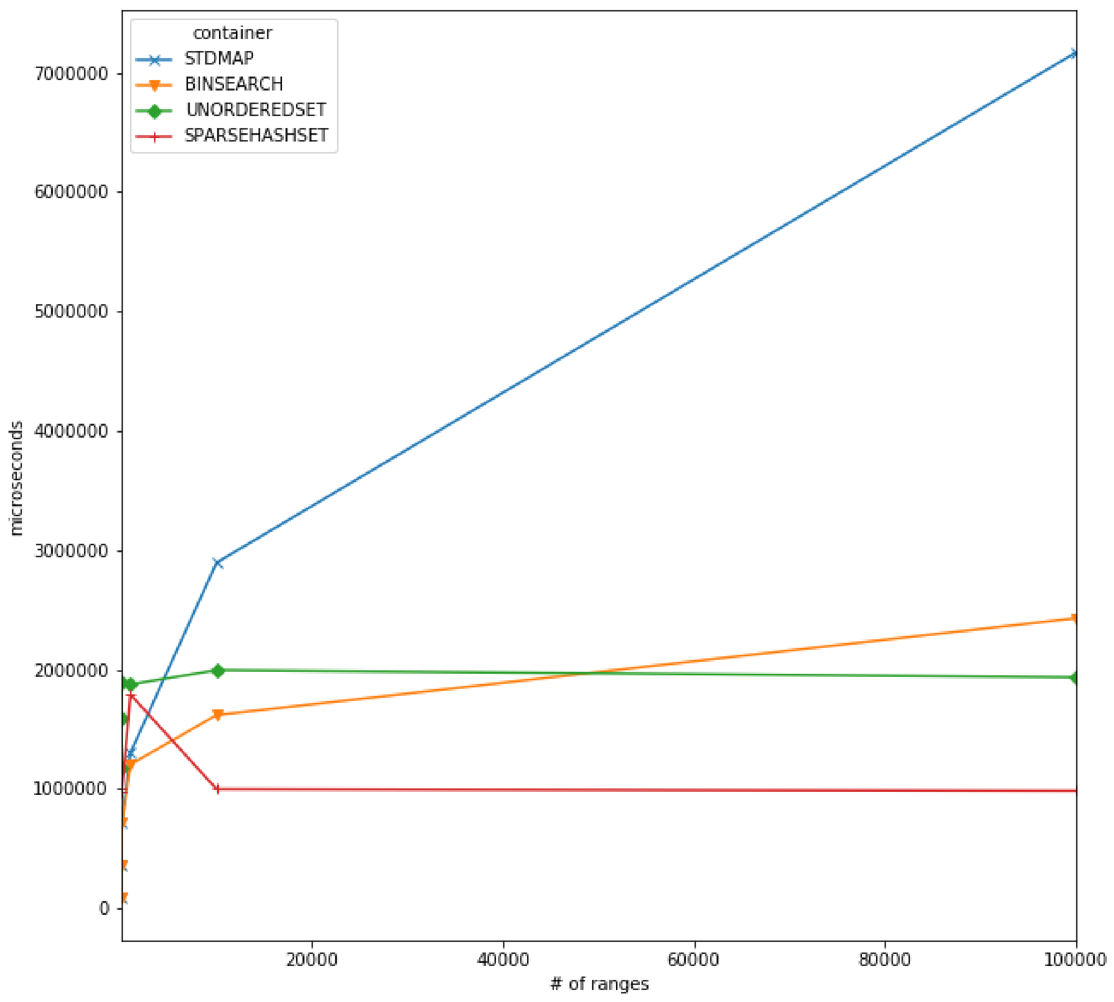
```
[10]:                STDMAP  BINSEARCH  UNORDEREDSET  SPARSEHASHSET
      ranges
      1.0           88722.0    85786.0     1176775.0       979977.0
      10.0         354061.0   356462.0     1587705.0       969146.0
      100.0        721478.0   715743.0     1887644.0      1003188.0
      1000.0      1304570.0  1204600.0     1874422.0      1788520.0
      10000.0     2894052.0  1617439.0     1993553.0       997317.0
      100000.0    7167739.0  2428427.0     1934161.0       983183.0
      1000000.0  18903723.0  5662575.0     2272937.0      1058766.0
```

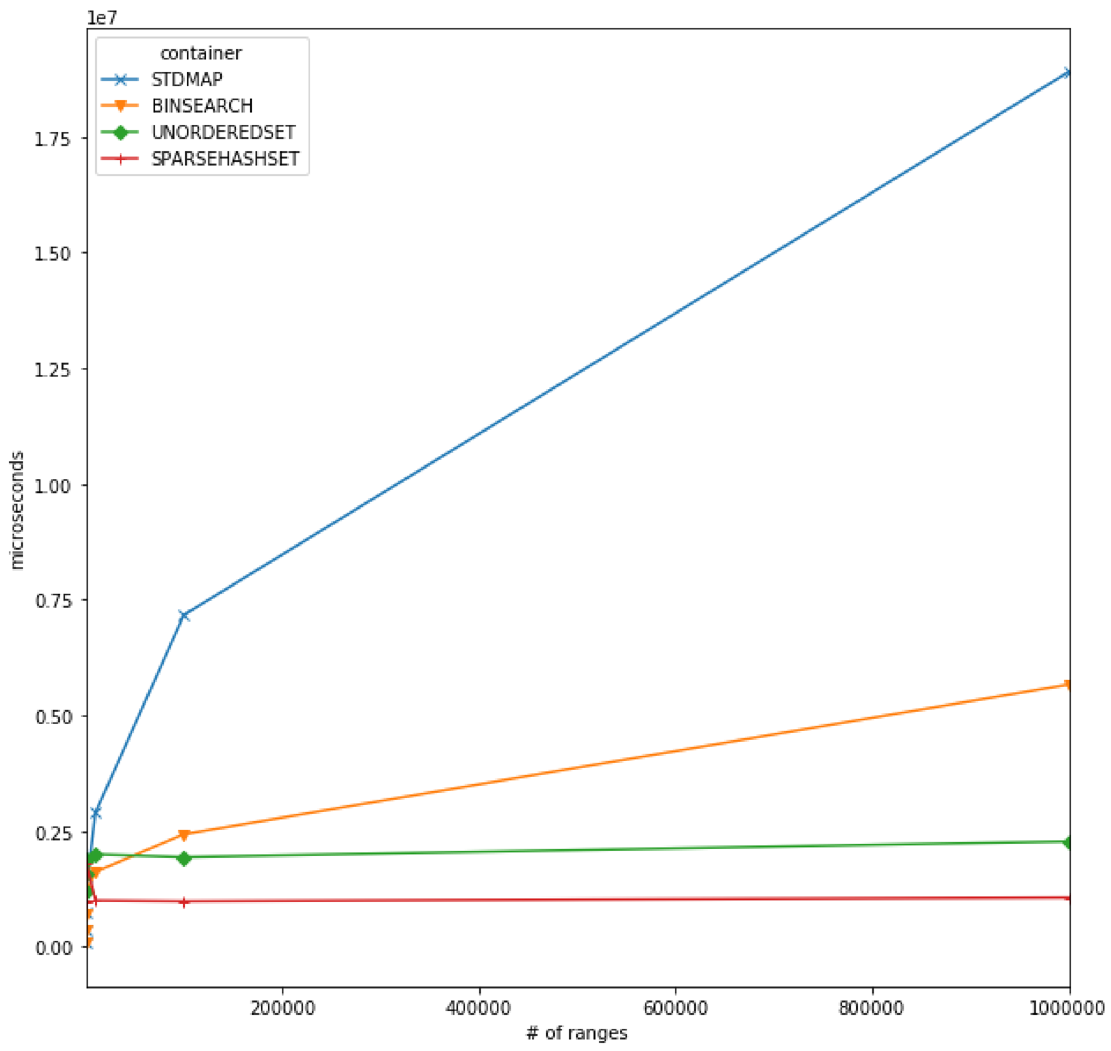
```
[14]: figsize_=(10,10)
      style_=['x-', 'v-', 'D-', '+-']
      frame.index.name = '# of ranges'
      frame.columns.name = 'container'
```

```
[15]: for lines in [3, 5, 6, 7]:  frame.head(lines).plot(figsize=figsize_,␣
       ↪style=style_).set(ylabel='microseconds')
```


container
STDMAP
BINSEARCH
UNORDEREDSET
SPARSEHASHSET
microseconds
0
250000
500000
750000
1000000
1250000
1500000
1750000
20
40
60
80
100
# of ranges


container
STDMAP
BINSEARCH
UNORDEREDSET
SPARSEHASHSET
microseconds
0
500000
1000000
1500000
2000000
2500000
3000000
# of ranges
2000
4000
6000
8000
10000


container
STDMAP
BINSEARCH
UNORDEREDSET
SPARSEHASHSET
microseconds
# of ranges
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
20000
40000
60000
80000
100000



Ip Ranges contains ~900k records. After creating ranges ~400k. It means, that we should choose between binary search (advantage - can handle increased number of ranges) and sparse hash.