Lab 3

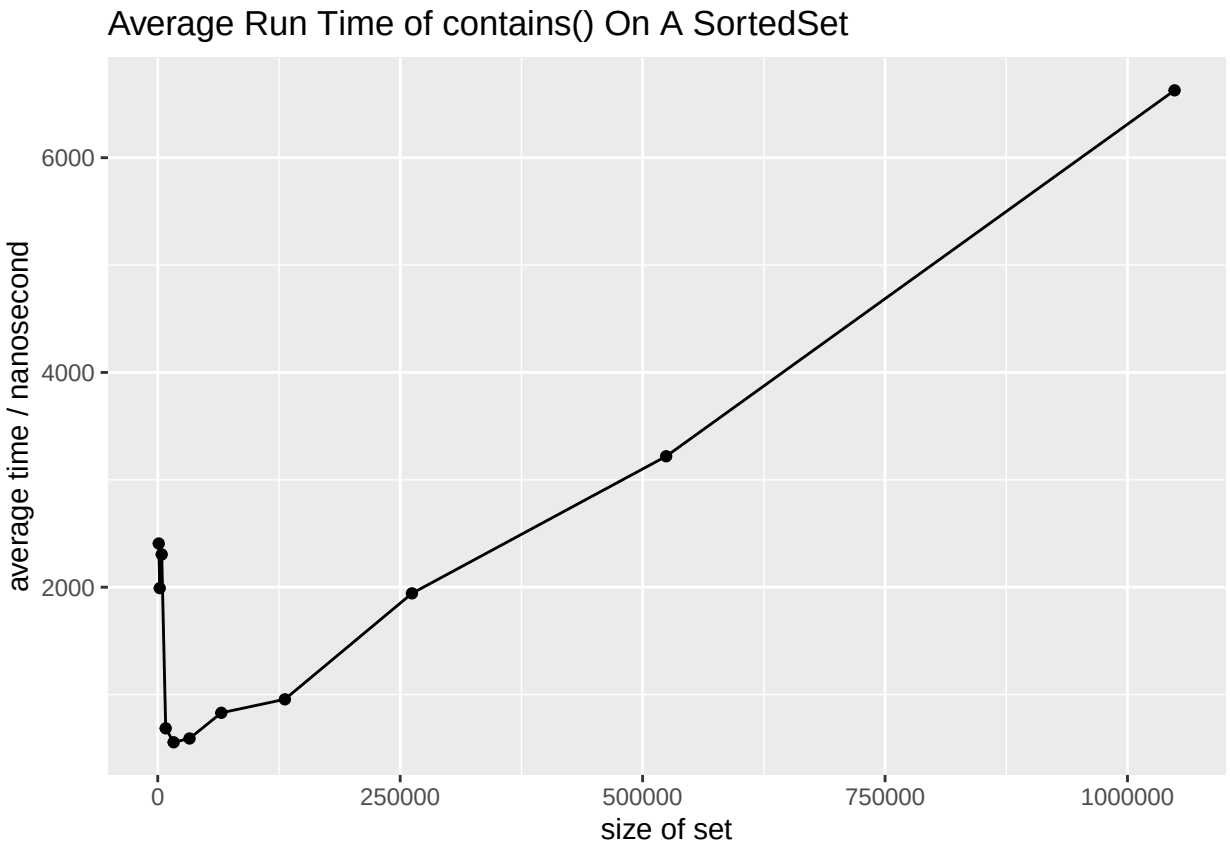
Muyuan Zhang

2022-11-11

Visualizing Timing Data

- Run the program and record the average time for `contains()` on 11 set sizes: 210, 211, 212... 219, 220.
- Time `contains()` several thousand times for each set size, and take the average.
- Plot these times against the set size.

```
library(ggplot2)
size <- c(1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576)
avgtime <- c(2406.64, 1991.2, 2305.01, 685.85, 555.42, 590.82, 830.0, 955.82, 1942.49, 3219.21, 6627.5)
timeingdata <- data.frame(size, avgtime)
ggplot(timeingdata, aes(size, avgtime)) + geom_line() + geom_point() + labs(
    x = "size of set",
    y = "average time / nanosecond",
      title = "Average Run Time of contains() On A SortedSet")
```



When the size goes above 16000, size-avgtime is almost linear. The left part of the graph is too crowded so we can try to apply a base-10 logarithmic transformation of x axis (size) as below.

```
ggplot(timeingdata, aes(size, avgtime)) + geom_line() + geom_point() + scale_x_log10() + labs(
    x = "size of set (base-10 log)",
    y = "average time / nanosecond",
      title = "Average Run Time of contains() On A SortedSet, x-axis Base-10 Log")
```


Average Run Time of contains() On A SortedSet, x−axis Base−10 Log
average time / nanosecond
6000
4000
2000
1e+03
1e+04
1e+05
1e+06
size of set (base−10 log)