Hypothesis. I expect the array to perform better than the binary tree for inputs of size less than 4096 (page size).

Methods. The experiment is ran on Ubuntu 19.04. GCC version is 8.3 and python version is 3.7.3. Also, matplotlib, pandas, and numpy are dependencies of plot.py

To run the experiment, first compile

```
g++ -fconcepts -std=c++2a -O3 code.cpp
```

Then execute

```
./a.out
```

which may take a few minutes to run. To plot the results use

```
python3 plot.py
```

The program compares sorted array and binary tree insertion performance for various input sizes n in the range [0, 1e4]. For each input size n, the program inserts n uniform random integers into a sorted array and into a binary tree. The total time taken for inserting n elements into each data structure is recorded.

Results.

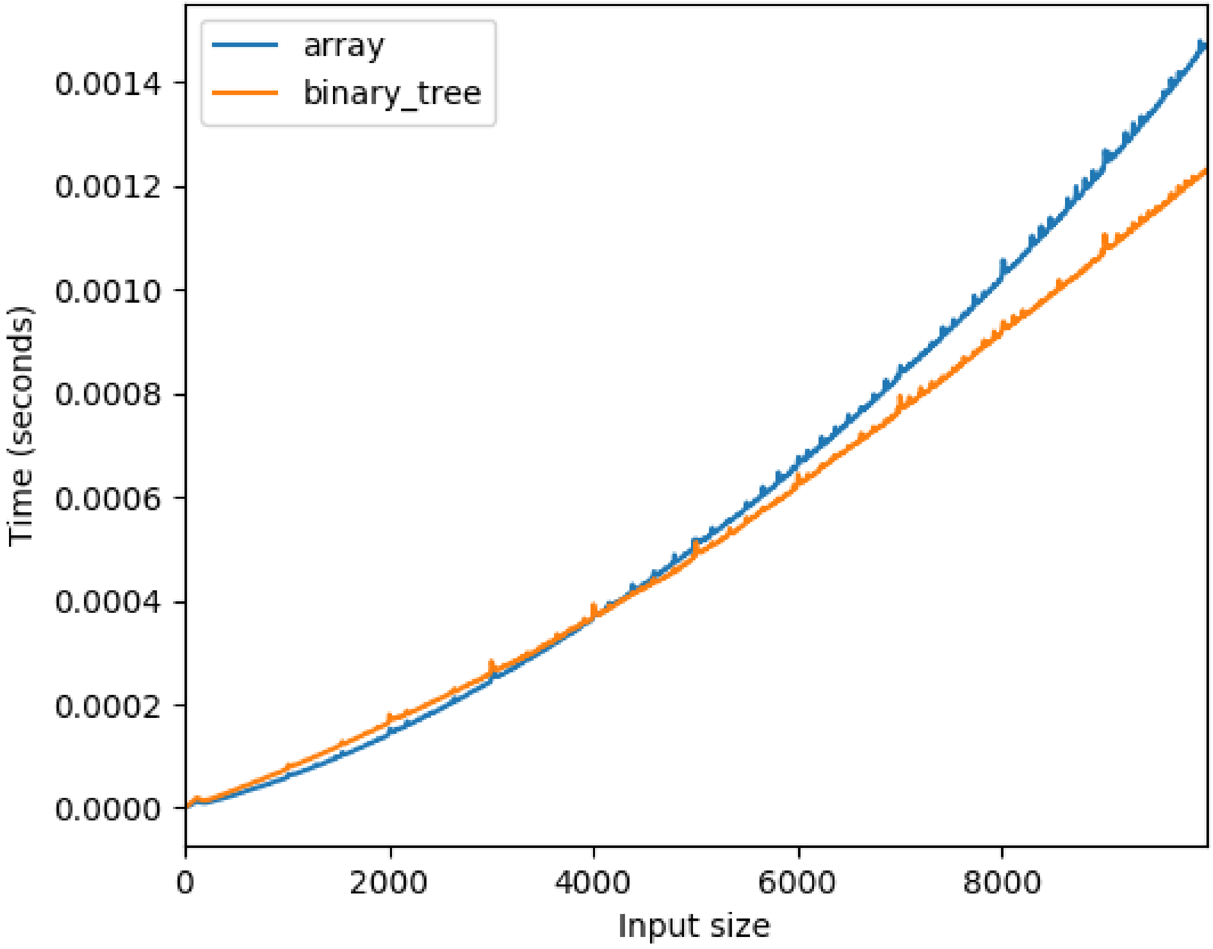

Figure 1

Figure 1 shows average time to insert n elements vs n.

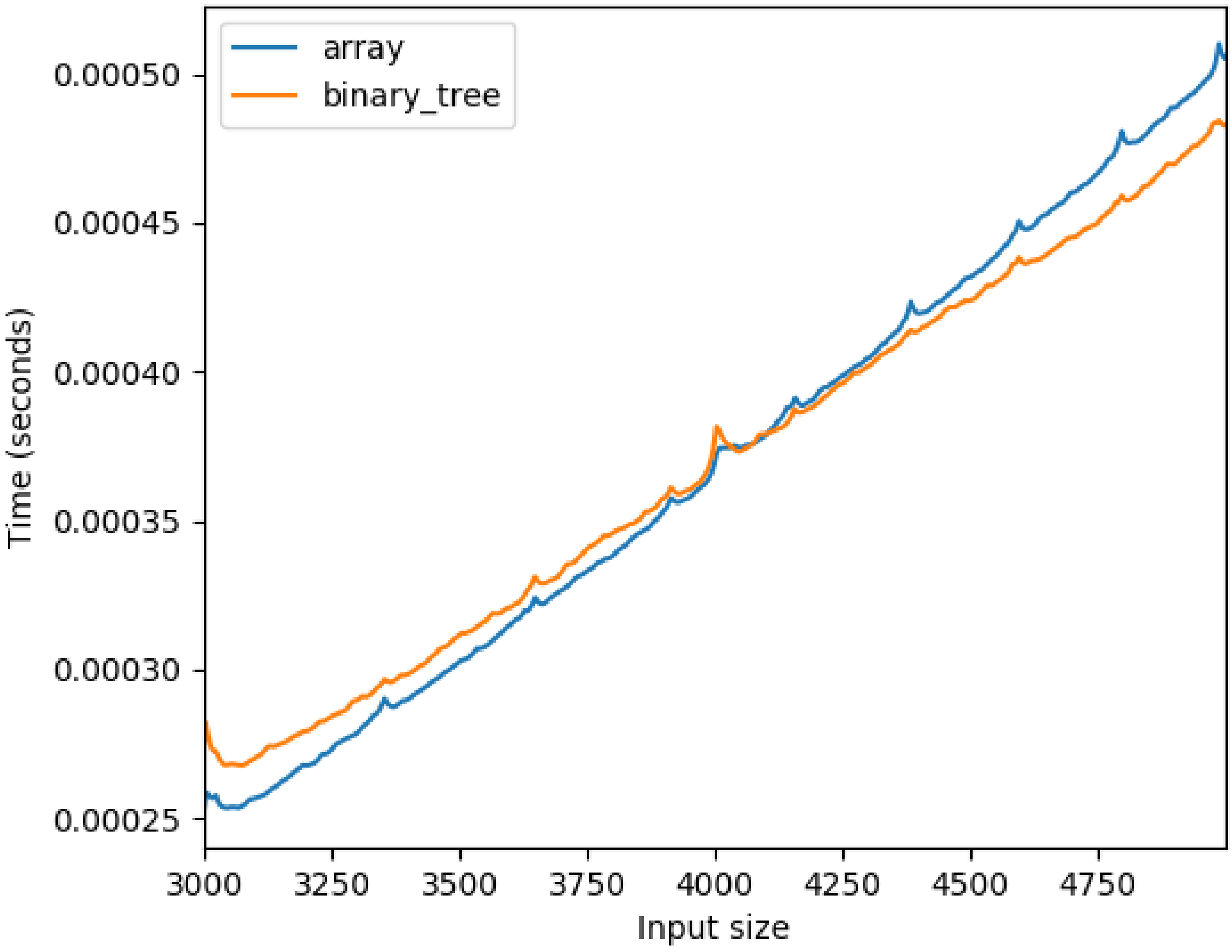

Figure 2

Figure 2 a zoomed in view of the previous plot to clearly show the intersection point.

Discussion. Figure 2 shows that the intersection point is roughly 4096 which is the page size on my machine.

Conclusion. The sorted array is a faster datastructure for maintaining relatively small sorted lists.