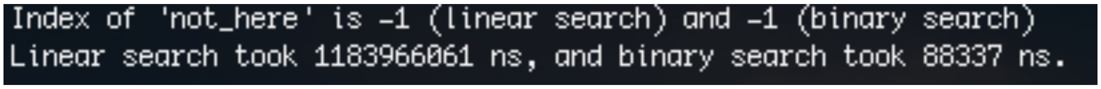
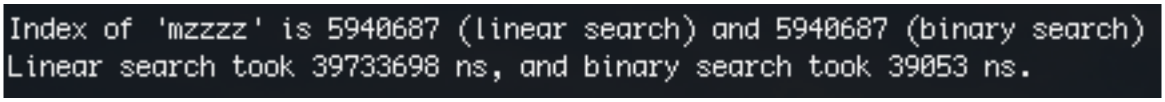
Caijun Qin
COP3502
11/06/2019

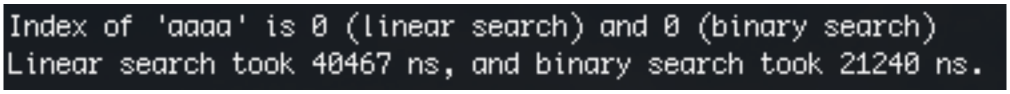
Lab 8 Report



```
Index of 'not_here' is -1 (linear search) and -1 (binary search)
Linear search took 1183966061 ns, and binary search took 88337 ns.
```



```
Index of 'mzzzz' is 5940687 (linear search) and 5940687 (binary search)
Linear search took 39733698 ns, and binary search took 39053 ns.
```



```
Index of 'aaaa' is 0 (linear search) and 0 (binary search)
Linear search took 40467 ns, and binary search took 21240 ns.
```

1) “not_here” is not even in the string array returned by getData(). Searching for this string results in checking all 17576 elements of the string array using linear search and approximately log$ 17576 checks using binary search. In each case, the maximum possible number of checks is used before arriving at the conclusion of “not_here” not being in the string array.

2) “mzzzz” falls near the middle of the string array returned by getData() from a birds’eye view of the extensively long array. Suppose that each first character leads a group of elements in the string array. “a****”, “b****”, …, “m****”, …, “z****”, where * is wildcard, each be a subarray of 26^4 strings starting with a common first character. From a big picture, “mzzzz” falls into the “m****” subarray, which is around the middle of the string array. Linear search therefore checks only about half of all elements.

3) “aaaaa” alphabetically is the very first element in the string array returned by getData(). This reduces linear search from linear time to constant time, if “aaaaa” is known beforehand to be searched. This is the best-case time complexity for linear search.

4) Worst-case time complexity for linear search is O(n) and O(log n) for binary search. Expectedly, binary search had a much shorter time period to complete than linear search for each test string.

5) Each iteration or checking some element in the string array for either search algorithms will not take exactly the same time. A very small amount of error occurs when counting time using System.nanoTime() because the machine architecture does not function perfectly with 100% accuracy. However, linear search takes way more iterations than binary search to find the same element. Linear search accumulates a greater error due to comparatively more iterations and spends more time to complete. Binary search minimizes the accumulation and chance to encounter more errors by finishing faster. Less error means closer time periods displayed even when searching for different strings.