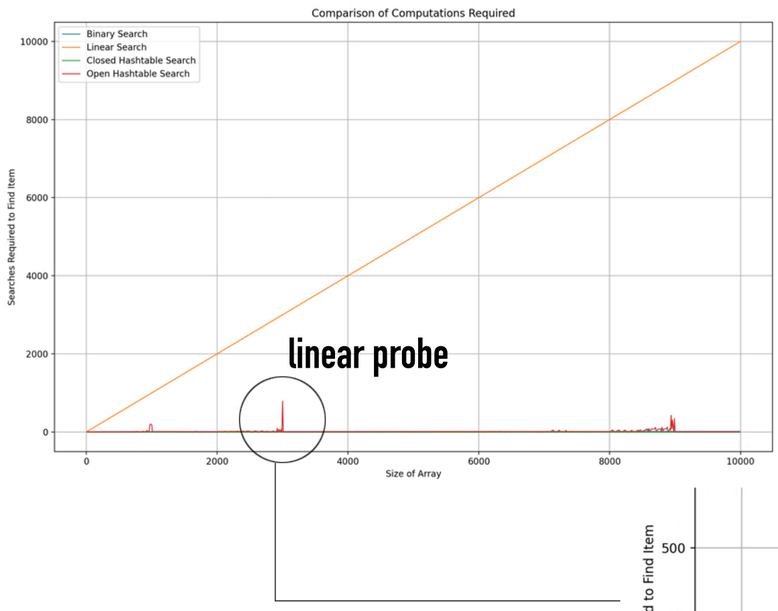
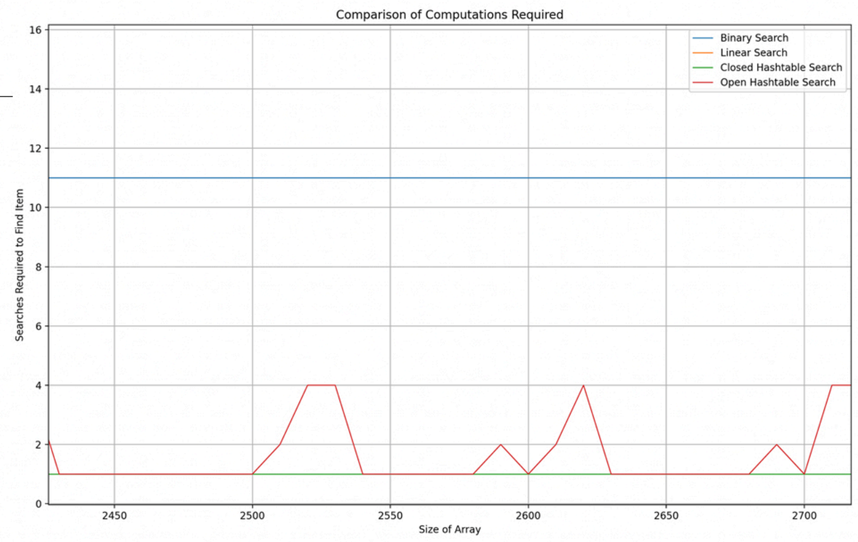
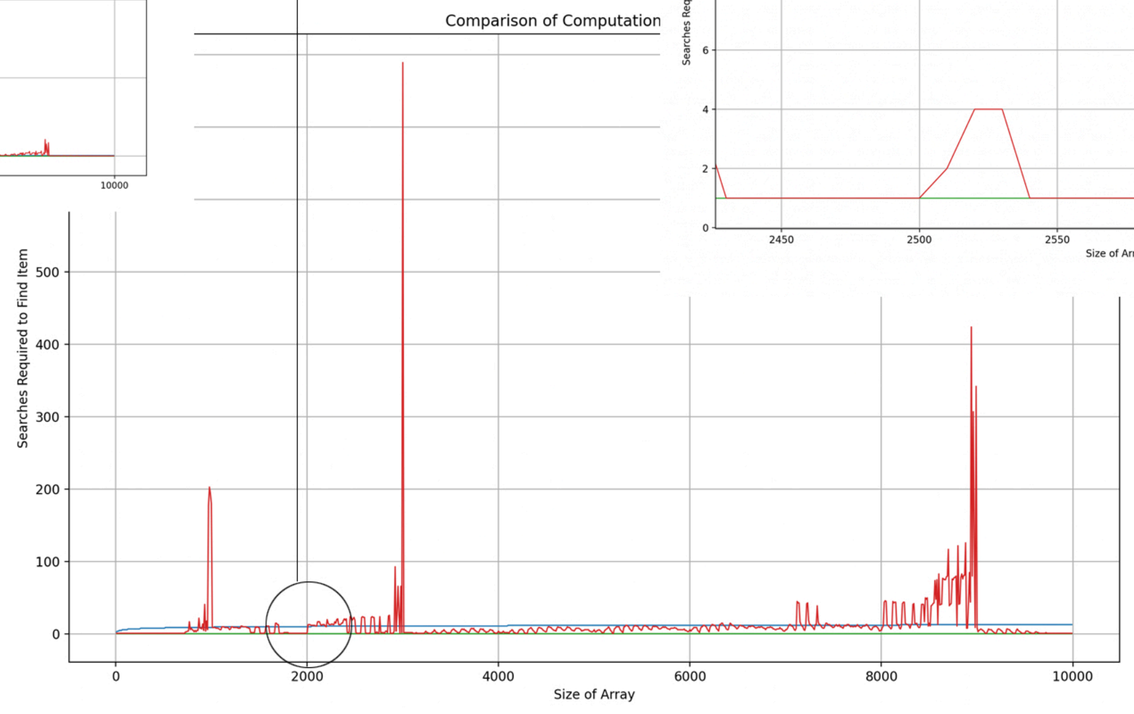


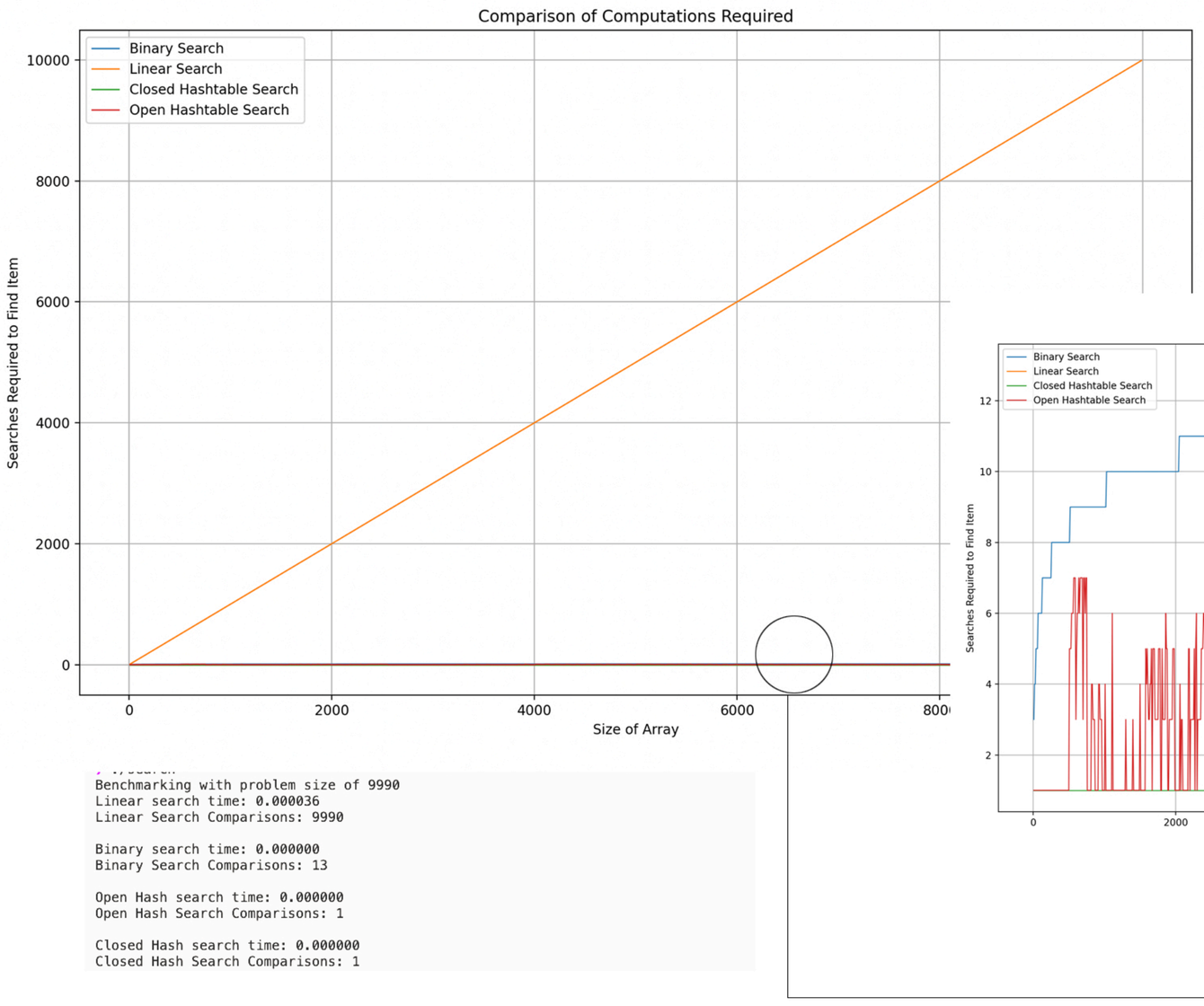





```
Benchmarking with problem size of 4990
Linear search time: 0.000022
Linear Search Comparisons: 4990

Binary search time: 0.000000
Binary Search Comparisons: 12

Open Hash search time: 0.000001
Open Hash Search Comparisons: 4

Closed Hash search time: 0.000001
Closed Hash Search Comparisons: 1
```

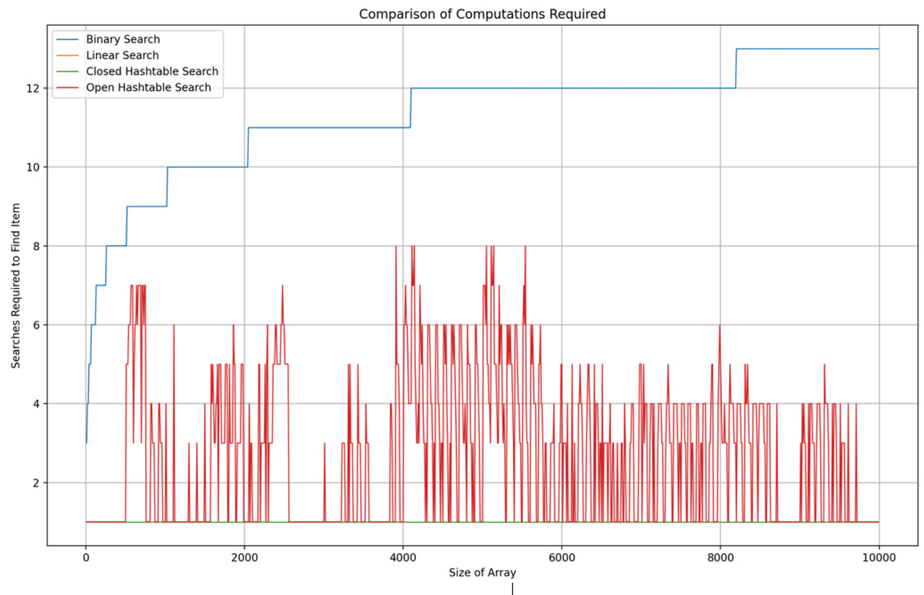
Item at end

When searching for an item at the end of an array, the linear search algorithm will always perform its worst at O(N) time, traversing the entirety of a list before finding its item. Binary search will perform at its average case, O(log(N)), as shown by its logarithmic curve here. Ideally, both maps will perform near O(1). Unfortunately, my open map's algorithm (red) seems to sometimes cluster particular sets of larger valued keys, in terms of hashing strings of digits that are longer. As shown in this graph, the open map has clusters around certain key values, degrading its search time to O(N) in those areas (though clearly still outperforming linear, even at its worst. The closed map continues to perform at near constant time.




```
Benchmarking with problem size of 9990
Linear search time: 0.000036
Linear Search Comparisons: 9990

Binary search time: 0.000000
Binary Search Comparisons: 13

Open Hash search time: 0.000000
Open Hash Search Comparisons: 1

Closed Hash search time: 0.000000
Closed Hash Search Comparisons: 1
```

quadratic probe



As seen here, quadtratic probing resloved the clustering and performance degredation issues with the open map over these hashed key values. This was the case for all value ranges (front, middle, back, item not found).