# Assignment Three

## Anthony Rodriguez

Anthony.rodriguez2@Marist.edu

October 31, 2020

## 1 Comparisons

| Linear Search | Binary Search | Hashtable Search |
|:---:|:---:|:---:|
| AVG: 337 | AVG: 17 | AVG: 11 |
| Total: 13444 | Total: 740 | Total: 501 |

## 2 Run Time

1. Linear Search [ BEST: $O(1)$, AVG: $O(n/2)$, WORST: $O(n)$ ]

   - The worst case of Linear Search can be explained due to the fact that n-elements in an array, with the most being the entirety of the array, would require sequential comparisons for the furthest element that may be searched. In other words, if the search is for the very last element, it would require an iteration through the array to the last index. Because there is no jumping or moving to other indices based upon current selection, the only direction to go is forward.

2. Binary Search [ BEST: $O(1)$, AVG: $O(logn)$, WORST: $O(logn)$ ]

   - Because binary doesn't require iteration through the entire array, this results in asymptotic runtimes fare better than linear. Unlike the former, there are several actions that can be taken during runtime. Due to binary dividing the given array into halves, it disregards large amounts of irrelevant data. In a few comparisons, the given array is shortened enough to increase the likelihood of finding the correct element. It follows the same principle of divide and conquer and utilizes subsets versus the whole.

3. Hash Table Search [ AVG: $O(1)$, WORST: $O(n)$ ]

   - The reason why Hash Tables experience their worse case is due to the potential of containing a large amount of chains. Though this can be argued that this is a rare phenomenon. In this case, since only a few may chain, Hash Tables experience constant time complexity and runtime.