# CSC236 Name: Tirtha Subedi

List the results you obtained for the three runs of linear search and sort + binary search with the time.time() function using war\_and\_peace.txt. Reflect on these results and whether or not they are what you expected.

|  |
| --- |
| Overall, they went well, but my binary search keeps returning a wrong number of a word. For example, when I try to return the word “nation” which is on the list for five times, I kept getting three or four or six. It was confusing but I finally figured it out. I was expecting to get the same number when put my word both linear and binary search but I only got five from the linear one but not the binary search code.  List of time when searching are:  Linear count is 27  The total linear search time is 0.01562666893005371 second  Binary search is 27  Time before sort 0.48442983627319336 seconds and the time after sorting is 0.0 seconds |

Binary search requires that the list of words be sorted before it can work, and the sorting algorithm itself is not fast if the list is large. Its execution time can typically vary between ϴ(n2) and ϴ(n log(n)), which is often higher than the performance of linear search, which is ϴ(n).

When would the fact that the list of words must be sorted before using binary search be a disadvantage?

|  |
| --- |
| It will definitely be a problem if you sorted the words and use linear search because it will take a long time find the item you are looking for. For example, what about if you looking through a list of thousands of words and the word you are looking for is the last line of the list of words, so linear does not work with the sorted lists. It will be problematic if you try to use linear search, you might find the item but it will definitely take a good amount of that you do not want to wait for your code to run. |

In what situations can you reduce the time cost of sorting to the performance of binary search?

|  |
| --- |
| I think you can reduce the time if you look through the list that contains less number of items because in that case, you do not have to go through the big list of words until you find your target. If you also implement your algorithm effectively because the more your algorithm is sorted well the better your code runs successfully. |

Suppose you decided to use both the linear and binary searches onto a sorted list to get them on an even playing field, so to speak. When would linear search always outperform binary search? When would binary search performance be faster? Explain.

|  |
| --- |
| The Linear search will outperform whenever the list of input is smaller because linear search is good and faster when you have a small number of the list. But if you have a large number of list, in that case, binary search beats the linear search because it takes a long time to go through every single item until you find the word or item you were looking for. |