Amandeep Singh

914814544

Frid, Yelena

ECS 036C A03

Report on P1

For our first programming assignment we were tasked to implement binary search and linear search to count the number of occurrences of fulfilled book requests. In my implementation what differs from a normal linear or binary search is that there is an overlapping loop going through the requested list and for each one of those elements the search is conducted.

So for linear the complexity turns out to be O(n^2) and for binary it is O(nlog(n)) both having their own respective pro’s and con’s associated with their implementations.

**Linear:**

Pros:

- Easy to implement

- Good for small search scales

- Isn’t required to be sorted before, which costs binary search a lot of time making linear faster than binary for smaller inputs

Cons:

- Can get costly for higher number’s

- Does not have direct access to data

**Binary:**

Pros:

- Standard search algorithm used for many scenarios

- Better big-o notation meaning much better performances for bigger number of inputs

- Fairly simple to implement.

Cons:

- Overkill for short datasets, actually making it more costly than linear for the same smaller size.

- Requires the list to be sorted which can add to complexity and slows down performance, this isn’t feasible for lists edited real time.

**Conclusion:**

Depending on the scenario both have different uses since they offer different things, mapping the data for this given program highlighted than linear worked better for the most part until the N’s got extremely large shown below:

|  |  |  |
| --- | --- | --- |
| N | Linear (avg ticks) | Binary (avg ticks) |
| 10 | 22.5 | 28.7 |
| 100 | 363 | 438 |
| 10,000 | 338,888 | 380,750 |
| 100,000 | 2 mins | 0.7 (s) |

