

Problem 1

a) [2 points if correct]

$O(n)$

b) [2 points if correct]

$O(n)$

c) [2 points if explanation is valid]

Sorting does not affect the worst-case runtime because you still have to potentially search the entire list linearly.

Problem 2

a) [2 points for explanation that somehow points out that we only gain something from hints if we know the list is sorted]

If the list were not sorted we would not know what direction to go from the hint. We also would still have to check every element potentially because we would never reach a point where we knew we could stop early.

b) [2 points if correct]

$O(1)$

c) [4 points if correct]

$O(n)$

d) [4 points for correct constant factor. 2 points if only give a reasonable explanation without the specific constant factor]

The search with this hint system would be roughly 26 times faster (only have to search through one letters worth of entries)

Problem 3

a) [2 points for a valid explanation pointing out that we don't have to search the whole list to find duplicates, or mentioning that there is no guarantee the duplicates should be placed together if unsorted]

Without sorting, we would have to always search the entire list to know whether there was a duplicate. Alternatively, if we weren't storing in sorted order, and wanted to store in insertion order, then we couldn't necessarily store duplicates in the same node because that would depend on insertion order.

b) [2 points for correct runtime, 2 points for stating there is a difference, 1 point for explanation. Half credit if answer assume n is number of nodes, but then answers everything correct w.r.t. that incorrect assumption]

$O(1)$. Yes, if we created a new node for every element, the search time would be $O(n)$. In the first case, there will no more than 100 nodes. So no matter the number of elements, we will have to check 100 nodes (a constant number of nodes). If there is instead 1 element per node, then as the list grows in size, the time to search grows also.

c) [2 points for correct runtime, 2 points for stating there is not a difference, 1 point for explanation]

$O(n)$. No it is not asymptotically different. Because there are an infinite number of names, then as we add more elements, the number of nodes in our list will grow, and therefore the time to search will also grow. While some elements may be duplicated, the number of nodes will still be proportional to n (in the worst case).