Assignment1 Analysis:

Task:1

In this ask, it's asked to sort the list of items in T(mn)complexity where m is the length of the word with maximum item and n is the length of list. So, the list data has been manipulated in the reading() function by replacing any space by putting words on the new line and separating two words by colon to make the difference between index and the song words. So, it's coming from the file format of example_song.txt file and the word has been given into new line and manipulation has been done according to the implementation. The common algorithm of counting sort is used to count the number of elements in order to sort it further using radix sort. It is to be stated that thw words are sorted using radix sort and put into new file called sorted_words.txt file to be calleted in the later task. In the radix sort algorithm, the index, list and size of the list is taken as input and using ordinary function the list is being sorted.

Task:2

In this task, the duplicate words are being removed as there can be a few repeating lines in the sorted file. They are definitely sorted by there are some duplicates. So, to remove those duplicates, the unique function u() is used to remove any duplicates and by using collate() function, it's used to replace the new line with space to use it in a proper way and thus we can use the list to manipulate on data.

Task3:

According to the given file which is unique sorted words so, here for the q number of queries to search in the sorted the file. The binary search would be the best to search through a sorted array. It has the complexity of O(log(U))*M where U is the

number of words and M is the longest word. In the case of the file, U would be the number of lines. Searching for q number of queries would take O(q * M(log U)) complexity. At the end to writing the p numbers of the ids for the respective queries would cost O(P). Hence overall complexity cost would be the O(q * M(log U) + p)