MH1403 Algorithms and Computing
Lab 4 Sorting Algorithms
(Week 12, 03.04.2023 – 07.04.2023)

Submission Instructions:

1. This lab is 4% of the final grade of this course.
2. The submission deadline is 11:59PM, 10 April (Monday).
3. You need to submit the codes of Task 1 and Task 2 through NTULearn.

Task 1. (2 marks) You will implement the merge sort function mergeSort(). The Python built-in sorting function/method should not be used in this task.

The following code is provided in the file task1.py (provided together with this lab document):

In the file malayenglish.txt (provided together with this document), each line contains a Malay word and its English translation. The Malay words are not sorted. Read the file into a list malayEnglishList (each element of malayEnglishList is a string consists of a Malay word and its English translation).

Call the function mergeSort() to sort the list malayEnglishList, then write the sorted list into malayenglish_sorted.txt . Each line in the file malayenglish_sorted.txt is a Malay word and its English translation.

Submit your code in the file task1.py . Please note that you ONLY need to implement merge sort in this task. When you are writing your code, you need to put task1.py and malayenglish.txt into the same directory so that the file malayenglish.txt can be read in task1.py .

Task 2. (2 marks) You will implement the binary search function binarySearch().

The following code is provided in the file task2.py (provided together with this lab document):

In the file malayenglish.txt (provided together with this document), each line contains a Malay word and its English translation. The Malay words are not sorted. Read the file into a list malayEnglishList (each element of malayEnglishList is a list with two elements: a Malay word and its English translation). The Python built-in sort method is applied to sort malayEnglishList according to the Malay words.

The program asks the user to input a Malay word, then calls the function binarySearch() to find its English translation from the sorted list malayEnglishList.

Submit your code in the file task2.py . Please note that you ONLY need to write the functions binarySearch() in this task. When you are writing your code, you need to put task2.py and malayenglish.txt into the same directory so that the file malayenglish.txt can be read in task2.py . The driver code provided in task2.py is for iterative implementation of binary search. If you use recursive approach to implement the binary search, you need to modify the driver code accordingly.

Some sample inputs/outputs are given below:

```
Enter a Malay word, or enter exit: abad
abad: century

Enter a Malay word, or enter exit: batu
batu: stone; rock

Enter a Malay word, or enter exit: panjang
panjang: long

Enter a Malay word, or enter exit: jialat
Cannot find the Malay word jialat

Enter a Malay word, or enter exit: exit
Program exits.
```

Remarks. In Lab 3, we used binary search tree to find the English translation of a Malay word efficiently. In this task, we apply binary search to a sorted list to translate a Malay word efficiently. The Malay-English dictionary remains static, so both approaches are efficient.