Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

4. [8 points] *Linear time sorting.*

   Suppose you are given an array of of strings, where different strings may have different numbers of characters, but the total number of characters over all strings is n. Prove that you can sort the strings in $O(n)$ time. The desired order is the standard lexicographic order, e.g., $\mathsf{a} < \mathsf{ab} < \mathsf{b}$.

Answer:

Let **m** be the number of strings, **k** be the range in which the characters within the strings lie (i.e. 128 for ASCII, a constant) and **L** be the length of the largest string present in the array.
Here L is the maximum number of characters in some string s_m.
As k is the range of characters (and a constant), it will be asymptotically smaller than n.

It is given is that **n** is the total number of characters over all m strings in the array.

The algorithm to sort the strings in lexicographical order is as follows:

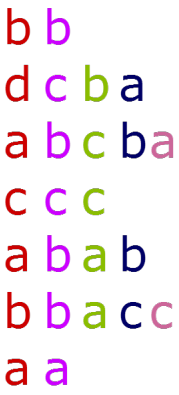
```
LexicographicSort(S)

        ▷ Create a hash map where key denotes string length
        ▷ value denotes index(s) of the corresponding strings
        L ← 0
        for i ← 1 to m, do                                          # O(m)
                ▷ Append operation will take constant time
                hashmap_append(length of s_i, i)
                L ← max(L, length of s_i)

        ▷ Let T be the set of indices of strings to be sorted at the i^th stage
        T ← φ
        for i ← L downto 1, do                                      # runs L times
                ▷ pops all the indices present at i^th key
                T ← T ∪ hashmap_pop(i)
                ▷ Sorts i^th component of strings in set T
                ▷ Sorts exactly the total number of characters (n) in total
                StableSort(T, i)                                    # O(m+k)
end
```

In the algorithm, we are using *Stable sort* for the i^{th} component on the strings given by the indices in the set T.

From the figure shown below, we can see that at the i^{th} stage, only those strings having at least i characters will be in set T and will be sent to stable sort.



In the figure above, we have an array of 7 strings ($m = 7$). The maximum length of a string in the array is 5 ($L = 5$ for *abcba*). Total characters over all strings, $n = 25$.

To sort these lexicographically, we first create a hashmap, with $key(l)$ = lengths of strings and $values$ = indexes of strings with length l. Then at each i^{th} stage, where i goes from first character of string to last character (1 till L for longest string), a group of characters *(color coded in image above)* will be compared with each other, to sort the strings.
Overall, as we can see, every character goes through the stable sort once, and a total of n characters is passed overall ($7 + 7 + 5 + 4 + 2 = 25$ in our case).

Hence, we can say that (as the constant associated with *stable sort* will be asymptotically smaller in comparison to n), all of this will at least take some cn comparisons.
$\Rightarrow$ T(m) $\in$ O(n)