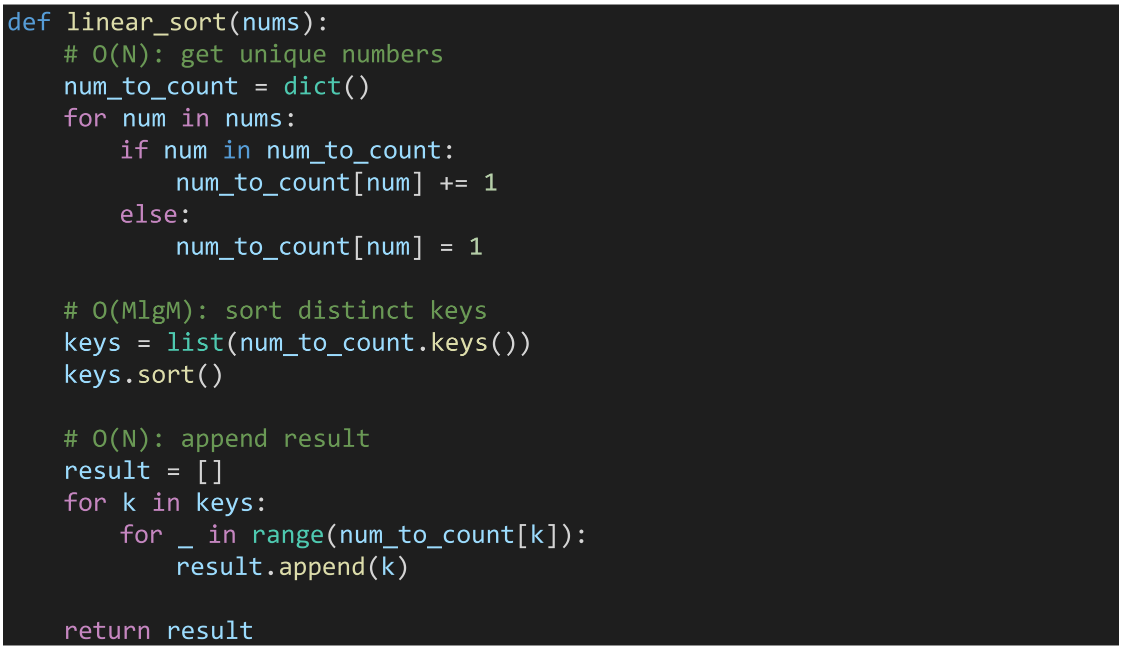
Problem Set 9

Daniel Wang (S01435533)

1. The idea is to retrieve distinct keys using a hashmap, where the key is unique number and value is its occurrence. Then, sort the keys in O(MlogM) time. Finally, append the result given the hashmap iterated though the sorted keys. The Python code is shown below:



```python
def linear_sort(nums):
    # O(N): get unique numbers
    num_to_count = dict()
    for num in nums:
        if num in num_to_count:
            num_to_count[num] += 1
        else:
            num_to_count[num] = 1

    # O(MlgM): sort distinct keys
    keys = list(num_to_count.keys())
    keys.sort()

    # O(N): append result
    result = []
    for k in keys:
        for _ in range(num_to_count[k]):
            result.append(k)

    return result
```

2. Assume the following lists is sorted from left to right. The traces would be:

(1) d = 1

pa	pe	of	th	th	th	ti	ai	al	no	fo	go	to	co	to	is

(2) d = 0

ai	al	co	fo	go	is	no	of	pa	pe	th	th	th	ti	to	to

3. I will first compute the maximum length in the list (assume it to be W). When backward the d from W-1 to 0, if current string at index d is empty, then its value will be put to the first position of radix (other characters' radices will be right-hand-side of this radix). Thus, string with empty character at index d will have the front of current list.

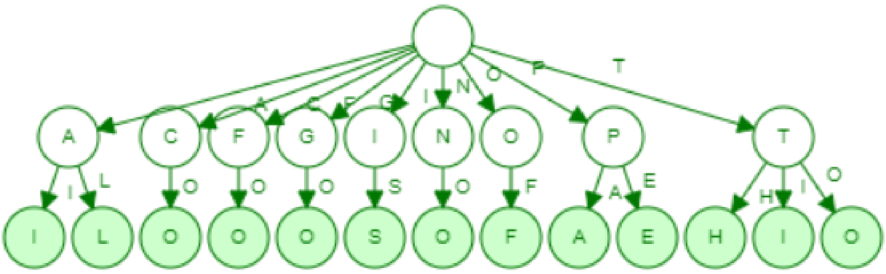
For example, if we have a string [“abc”, “ab”, “aba”], then:

(1) d=1, result = [“ab”, “aba”, “abc”] since the radix before prefix sum will be:

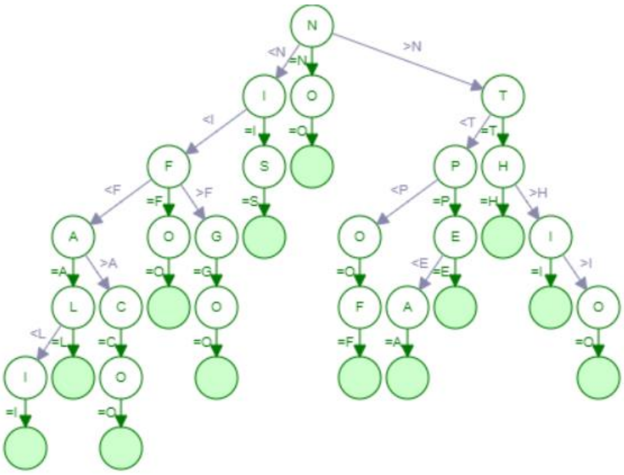
1	1	0	1
“_”	“a”	“b”	“c”

(2) d=0, order is not change, so output [“ab”, “aba”, “abc”]

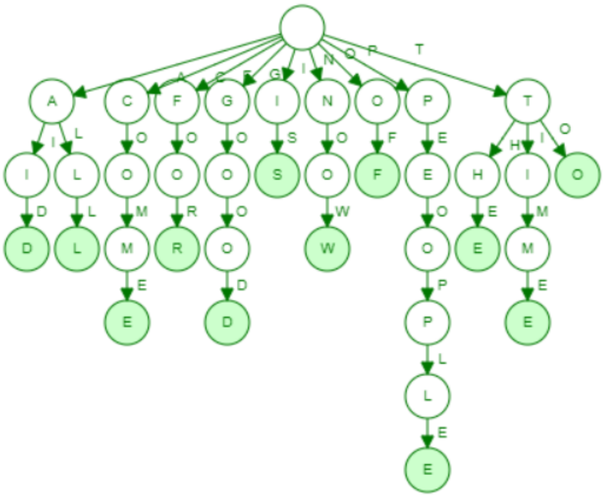
4. The result is shown as below:



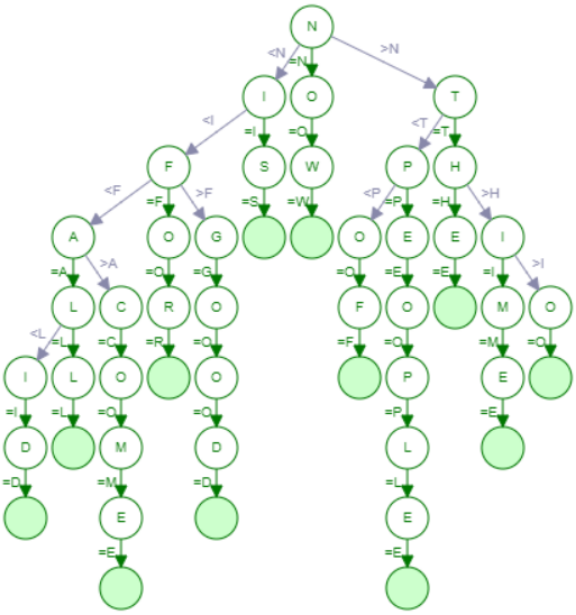
5. The result is shown as below:



6. The result is shown as below:



7. The result is shown as below:



8. Sub-questions are answered below, where boldface number represents end state.

(1) The transitions are:

Input		A	B	C	A	A	A	B	A	B	A	B	A	C	A	A	B	B
State	0	1	2	0	1	1	1	2	3	4	5	4	5	**6**	-	-	-	-

(2) The transitions are:

Input		A	B	C	A	A	A	B	A	A	B	A	C	A	A	B	B	A
State	0	1	2	0	1	1	1	2	3	1	2	3	0	1	1	2	0	1