1. My Python code:

```python
# Hsuan-You Lin Module 10 Problem Set Question 1.
def counting_Sort(arr):
    # creat the hashmap for array and count the elements
    HashMap = dict()
    output = [None] * len(arr)
    for i in range(len(arr)):
        num = arr[i]
        # if any element repeats itself,
        # simply increase its count
        if num not in HashMap:
            HashMap[num] = 0
        HashMap[num] += 1

    # sorted the hashmap by keys
    HashMap = dict(sorted(HashMap.items(), key = lambda item: item[0]))
    sorted_num = list(HashMap.keys())

    # cumulative count
    for i in range(1, len(sorted_num)):
        HashMap[sorted_num[i]] += HashMap[sorted_num[i - 1]]

    # Build the output array
    for num in arr:
        output[HashMap[num] - 1] = num
        HashMap[num] -= 1

    # Copy the output array to arr,
    # so that arr now contains sorted characters
    for i in range(0, len(arr)):
        arr[i] = output[i]
    return arr

if __name__ == "__main__" :
    arr = [4, 2, 2, 8, 3, 3, 1]
    print("Before sorted array is: ", arr)
    ans = counting_Sort(arr)
    print("After counting sorted array is: ", ans)
```

```
Module10 — -bash — 80×24
(base) pisces:Module10 pisces$ python Q1.py
Before sorted array is:  [4, 2, 2, 8, 3, 3, 1]
After counting sorted array is:  [1, 2, 2, 3, 3, 4, 8]
(base) pisces:Module10 pisces$
```

2. My Answer:

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

My steps:

Data	D=1	D=0
no	pa	ai
is	pe	al
th	of	co
ti	th	fo
fo	th	go
al	th	is
go	ti	no
pe	ai	of
to	al	pa
co	no	pe
to	fo	th
th	go	th
ai	to	th
of	co	ti
th	to	to
pa	is	to

3. My Answer:

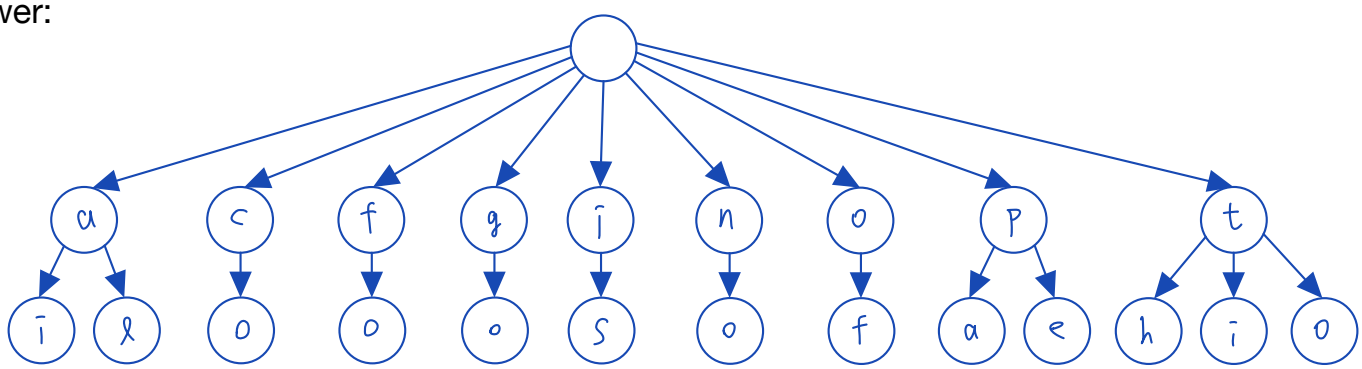
First, get the maximum lengh in the list, and do the radix sort, if index d is empty in the string, then put it to the front of the list, make sure the shortest string always in front of the list.

example:

Sort the following strings: "tho", "th", "the".

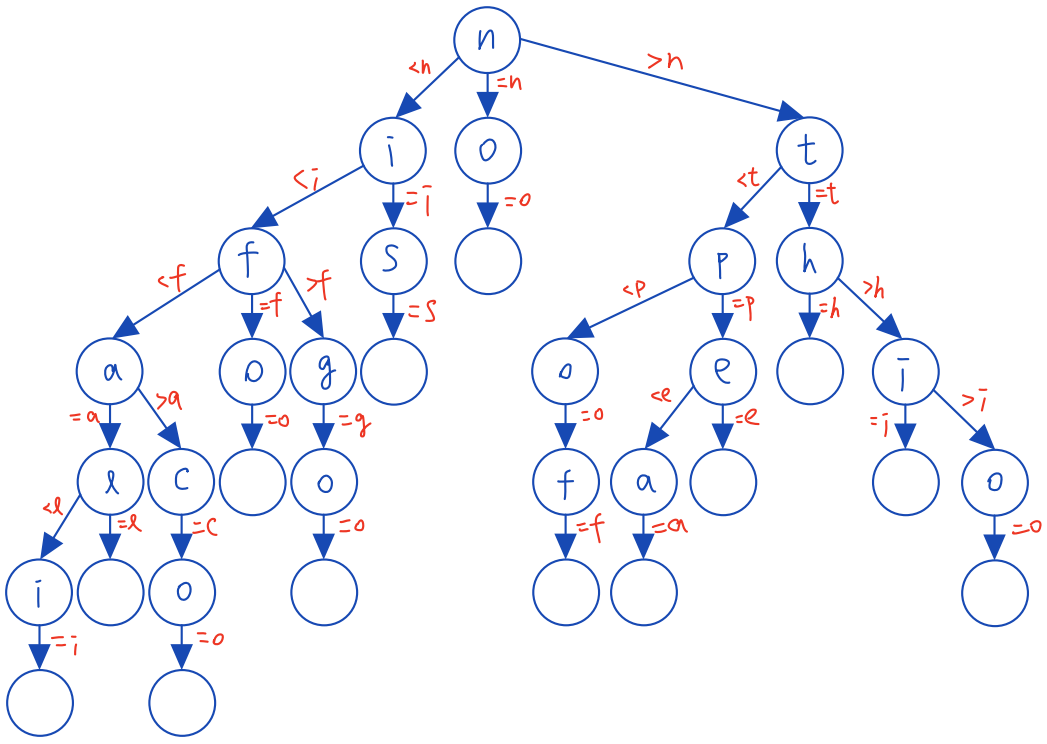
Data	D=2	D=1	D=0
tho	th	th	th
th	the	the	the
the	tho	tho	tho

4. My Answer:



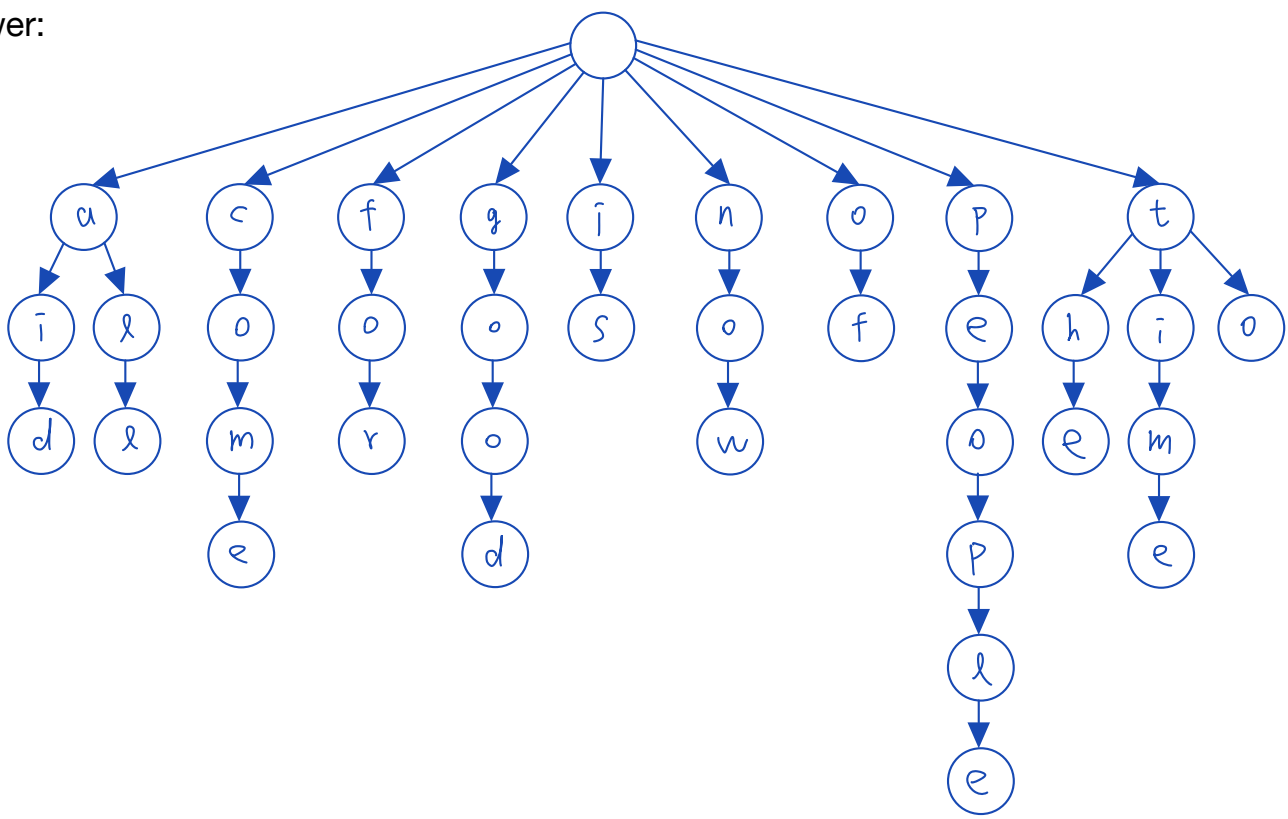
---

5. My Answer:

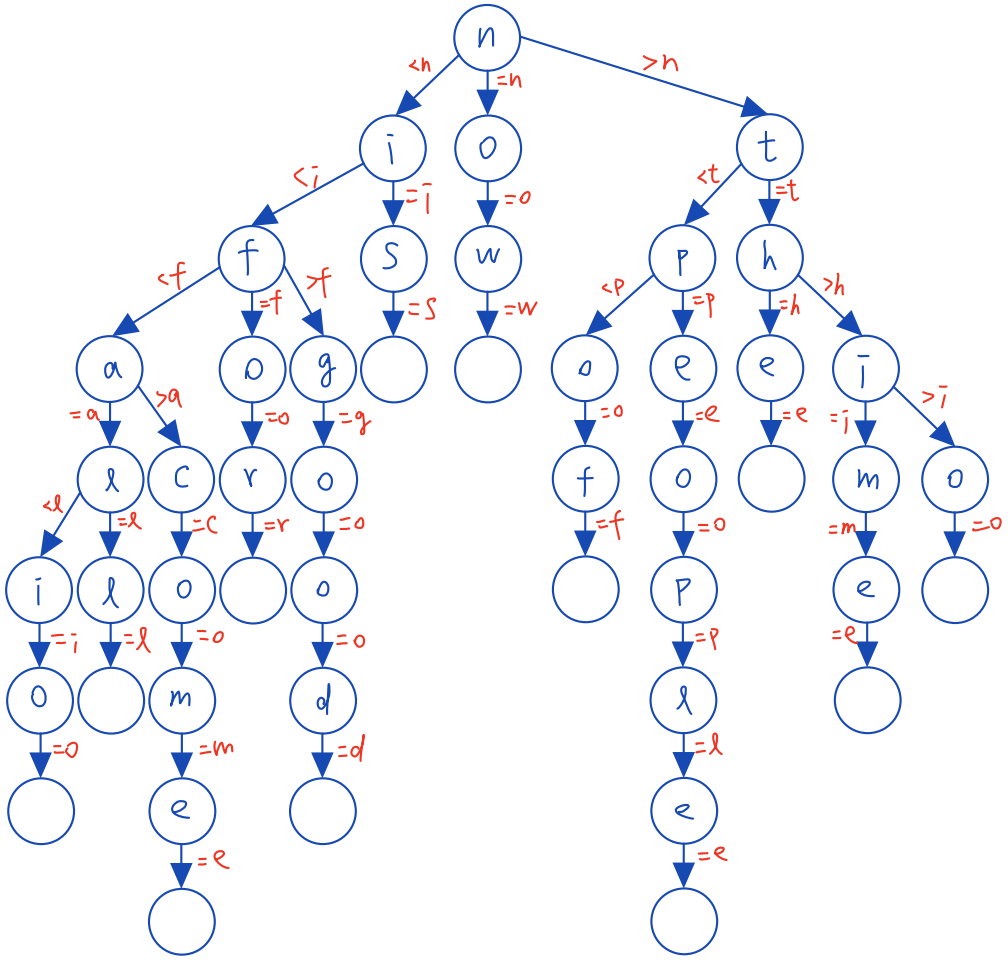


---

6. My Answer:



7. My Answer:



---

8. My Answer:

input data	state
	0
A	1
B	2
C	0
A	1
A	1
A	1
B	2
A	3
B	4
A	5
B	4
A	5
C	6
A	none
A	none
B	none
B	none

⇒

←end state

input data	state
	0
A	1
B	2
C	0
A	1
A	1
A	1
B	2
A	3
A	1
B	2
A	3
C	0
A	1
A	1
B	2
B	0
A	1