CMPE 343
Fall 2022
Programming Homework 4

This assignment is due by 23:55 on Friday 30, December 2022.

Homework Q&A Sessions:

- December 20, 19:00-20:00, Zoom ID: https://tedu.zoom.us/j/96597483976
- December 27, 19:00-20:00, Zoom ID: https://tedu.zoom.us/j/98012980010

Note: Please make sure that you have read the HW document well before participating. However, no HW related questions will be accepted except from the above options.

PROGRAMMING TASK

In this part, you must implement your own trie data structure by taking inspiration from your textbook and use it to help to solve problem. You are not allowed to use any external library or .jar file. Any solutions without using trie data structure are not evaluated!

Question 1 (25 points):

In this question, you should find the number that has minimum XOR result with the given mask.

Let's assume that we have following 9 bit binary numbers:

1 1 0 0 0 0 0 0 0

1 1 1 0 1 0 1 0 0

1 1 1 0 0 0 0 0 0

0 0 0 0 0 1 1 0 0

0 0 0 0 0 0 1 1 0

1 0 1 0 1 0 1 0 0

We want to find the number that has minimum XOR result with the given mask: 1 0 1 0 0 1 1 1 1, and the result should be: 1 0 1 0 1 0 1 0 0.

Example: 1 1 0 0 0 0 0 0 0 (first number above) (XOR) 1 0 1 0 0 1 1 1 1 (mask) = 0 1 1 0 0 1 1 1 1

Sample Run:

```
6                               // number of binary numbers
110000000                       // number 1
111010100                       // number 2
111000000                       // number 3
000001100                       // number 4
000000110                       // number 5
101010100                       // number 6
101001111                       // mask
101010100
```

```
5                               // number of binary numbers
110000000                       // number 1
111010100                       // number 2
111000000                       // number 3
000001100                       // number 4
000000110                       // number 5
101001111                       // mask
111000000
```

```
6                               // number of binary numbers
110000000                       // number 1
111010100                       // number 2
111000000                       // number 3
000001100                       // number 4
000000110                       // number 5
101010100                       // number 6
010101011                       // mask
000001100
```

You should use trie data structure to solve this question. Other solutions that do not use trie approach will not be evaluated and penalize with 0 point