

FIFO

front()

front

→ ~~4~~ ~~3~~ ~~1~~ ~~10~~ ~~2~~ ~~6~~ ~~1~~ ~~3~~ ~~1~~ ~~10~~ ~~2~~ ~~6~~

n=6

[6 | 2 | 10 | 1 | 3 | 4]

TC = O(n)

AS = O(n)

f
→ [4, 3, 1, 10, 2, 6]

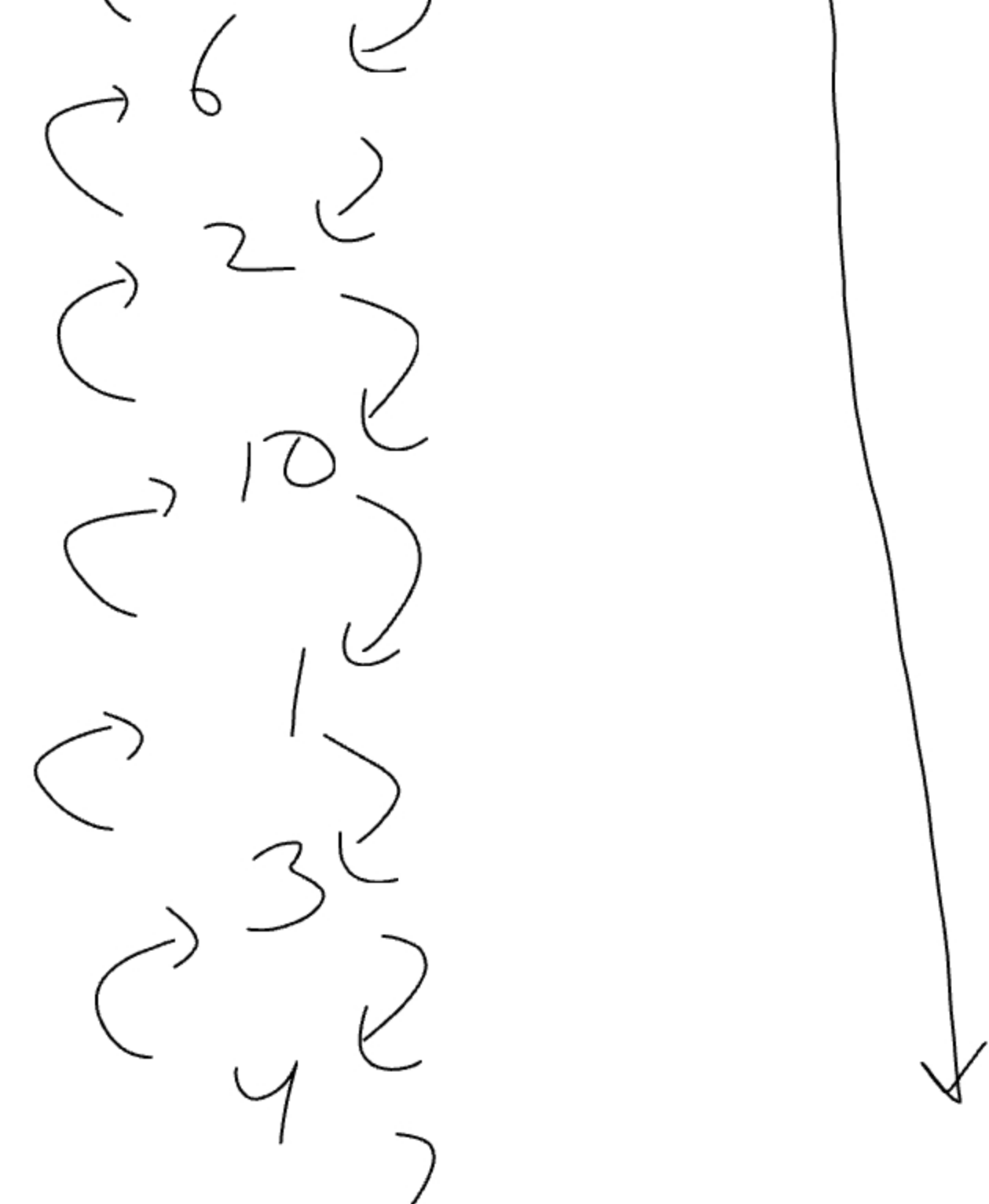
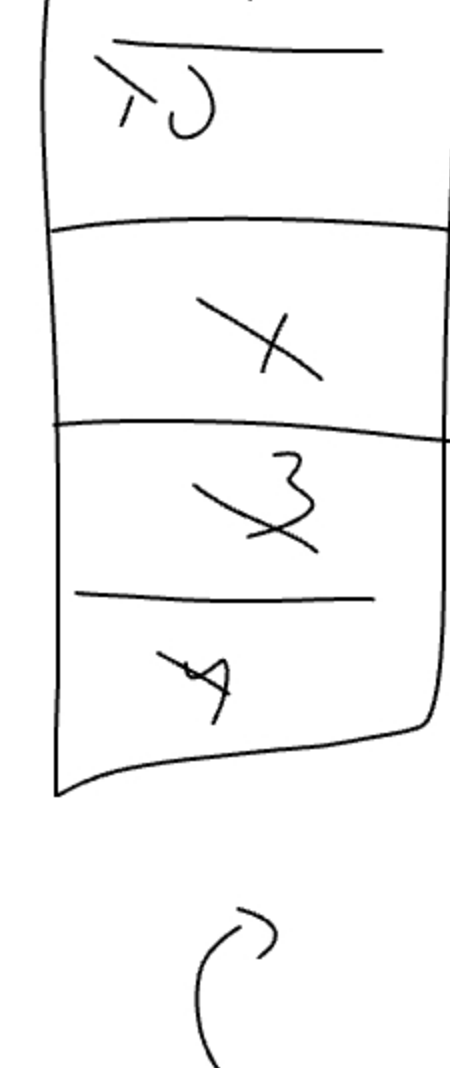
[6, 2, 10, 1, 3, 4]

[~~4~~, ~~3~~, ~~1~~, ~~10~~, ~~2~~, ~~6~~]

[6, 2, 10, 1, 3, 4]

TC: O(n)

AS: O(n)



0
→ 1
→ 10
→ 11
→ 100
101
110
111
1000
1001
→ 1010
1011
1100
1101
1110
1111

n=10

FIFO → Queue

→ 1010

TC: O(n * log(n))

n=10

Queue: ~~1~~, ~~10~~, ~~11~~, ~~100~~, ~~101~~, ~~110~~, ~~111~~, ~~1000~~, ~~1001~~, ~~1010~~, 1011

res: [1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010]

n=10

res: [1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010]

TC = O(n)

AS = O(1)

s = "abc"

aabc zz

i/p: aabc
o/p: a#bb

i/p: zz
o/p: z#

i/p: aabc
o/p: a#bb

freq
a → 2
b → 1
c → 1

i/p: aacb
o/p: a#cc

first 0a

i/p: aabc
o/p: a#bb

freq
a → 2
b → 1
c → 1

TC: O(n)

AS: O(1)

charTo11 = # b

i/p: aabbccdde
o/p: a#b#c#ddde

freq
a → 2
b → 2
c → 1
d → 3
e → 1

i/p: da

d

o/p: dd

freq
a → 2
b → 2
c → 1
d → 3
e → 1

i/p: ada
o/p: aad

dad

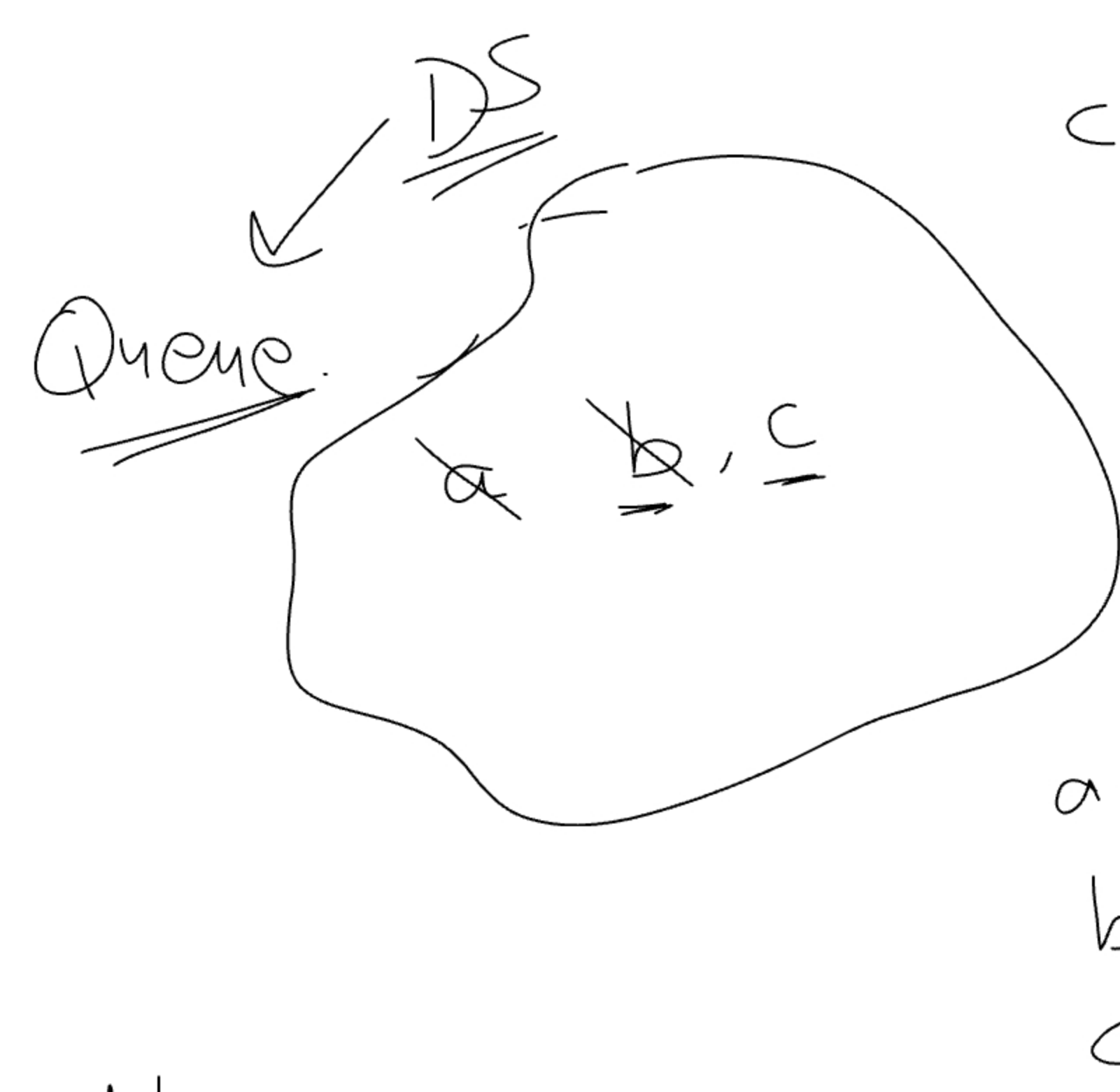
i/p: aabc
o/p: a#bb

a → 2
b → 1
c → 1

i/p: daabbc
o/p: d#a#bb

d → 2
a → 2
b → 1
c → 1

i/p: aabcb
o/p: a#bbcb



a → 2
b → 2
c → 1

i/p: aabccdba

```
string firstNonRepeatingCharsUsingQueue(string s) {
    vector<int> freq(26, 0);
    queue<char> q;
    string result = "";

    for (int i = 0; i < s.length(); i++) {
        int charIdx = (s[i] - 'a');
        freq[charIdx]++;

        if (freq[charIdx] == 1) {
            q.push(s[i]);
        }

        while (!q.empty() && freq[q.front()] > 1) {
            q.pop();
        }

        if (q.empty()) {
            result.push_back('#');
        } else {
            result.push_back(q.front());
        }
    }

    return result;
}
```

Queue: a

a → 1