



Minimum Window Substring

str1: ^{inc} d b a e c b b a b d c a a f b d d c a b g b a

str2: ^{orc} a b b c d c

Map'

d	→	2
b	→	3
a	→	4
f	→	1
c	→	1
g	→	1

Map2

a	→	1	✓
b	→	2	✓
c	→	2	✓
d	→	1	✓

26

my window
should have each
of them -

ans = c b b a b d c → 7

can I reduce
this effort?

$2 \times N \times 26$
↓
len(str1) = $O(N)$

str.substring(enc+1, next)

str1: dbaecbbabdc a a f b d d c a b g b a

str2: a b b c d c

a → 1
b → 2
c → 2
d → 1

d → 1
b → ~~1~~ 3
a → 1
c → 1

dmant = 6

mant = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

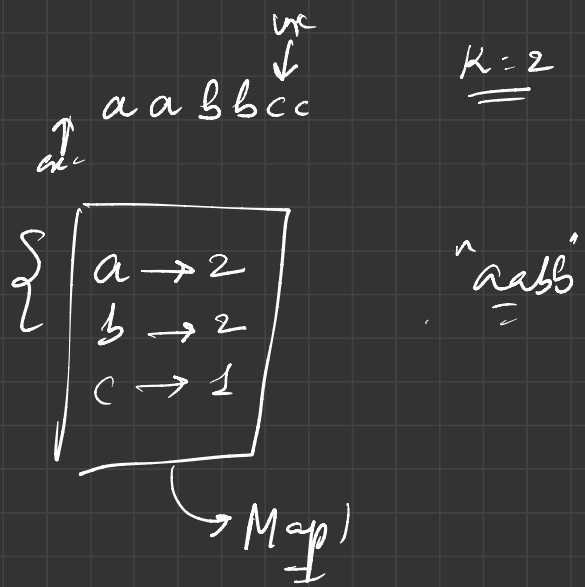
map2

$$O(2^N) \approx \boxed{\underline{O(N)}}$$

```

while (true) {
    boolean f1 = false;
    // include people
    while (inc + 1 < s.length() && map1.size() <= k) {
        inc = inc + 1;
        char ch = s.charAt(inc);
        map1.put(ch, map1.getOrDefault(ch, 0) + 1);
        if (map1.size() == k) {
            String pans = s.substring(exc + 1, inc + 1);
            if (ans.length() == 0 || pans.length() > ans.length()) {
                ans = pans;
            }
        }
        f1 = true;
    }
    boolean f2 = false;
    // exclude people
    while (exc < inc && map1.size() > k) {
        exc = exc + 1;
        char ch = s.charAt(exc);
        map1.put(ch, map1.get(ch) - 1);
        if (map1.get(ch) == 0) {
            map1.remove(ch);
        }
        f2 = true;
    }
    if (f1 == false && f2 == false) {
        break;
    }
}

```



} Break 2:10 pm }