Friday, 19 July 2024
8:45PM

20+ Q → Binary search ?? 20 / 23

Hashing 20 / 27
+8
32 / 35 → Hashing
↳ Friday ✓

≈ 20 + 15

+3

Friday 60
+ Stack
CW (??)
Friday

1.5 ✓

Contest
↳ 🙋 ✓

No
↳ TC ✓
SC × ✓

Edge cases
↳ Problem statement + Understand
↳ → What ?? → Understanding → 3

How?? ✓

Practice ✓

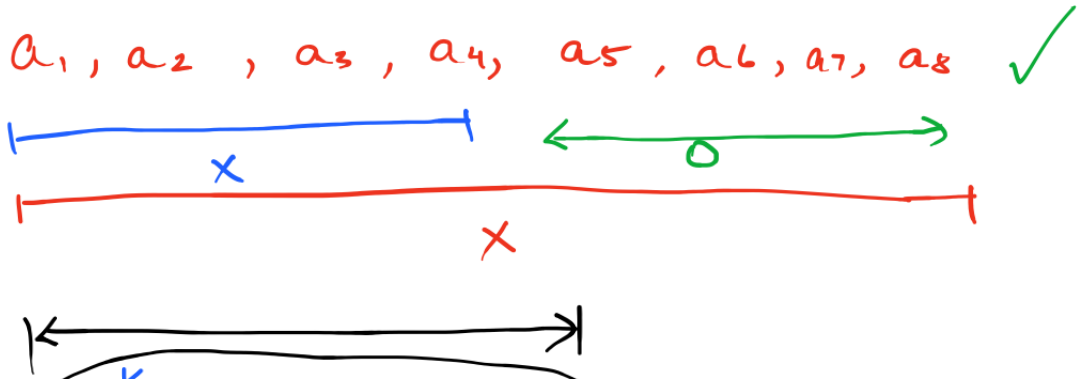
Day ✓

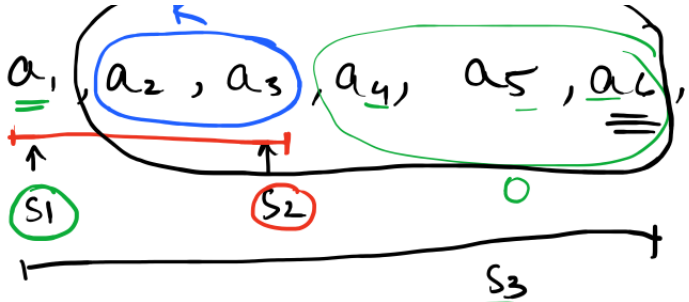
Questions

Subarray Sum equals K

Subarray with sum 0

return the greatest length of subarray having sum 0





K = 0

S_3

$S_2 = S_1 + K$

$S_1 + K + 0$

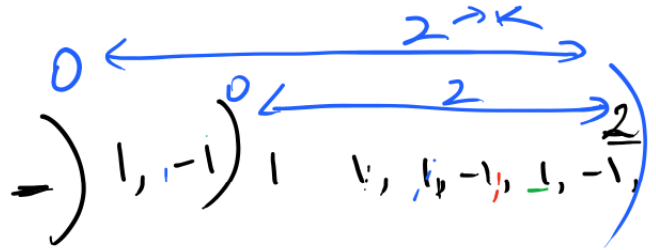
↳ $S_1 = S_2 - K$

↳ $S_1 = S_3 - K$

Sum − K

↳ If this value is already visited from index [0, a_i]

↳ a_{i+1} → current ⇒ K (sum)



K = 2 ✓

ans = 0 + 2 + 2 + 2 + 2 + 2

Sum = 0 + 1 − 1 + 1 = 1 + 1 = 2 + 1 = 3

2

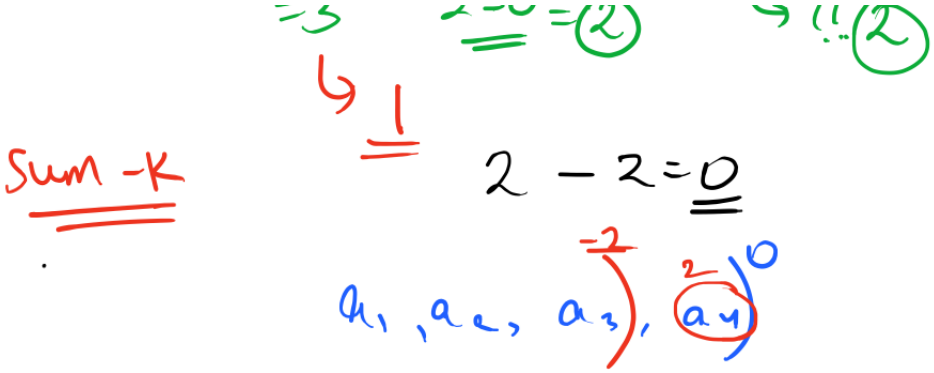
2 + 1

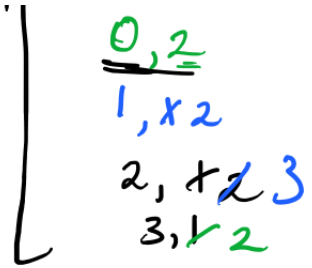
1

Hashmap

↳ sum vs count

Map





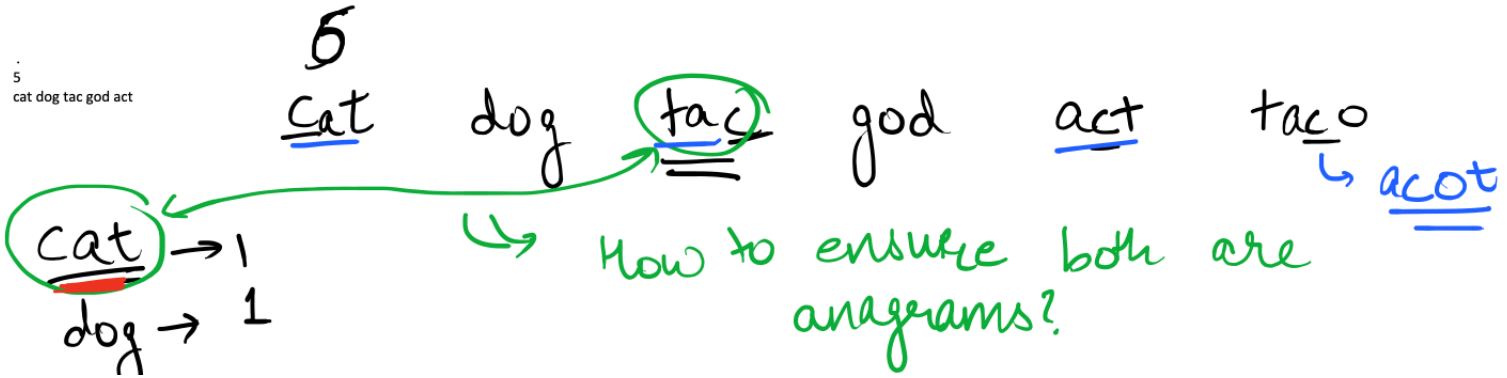
TC: O(n)

SC: O(n)

Group Anagrams

↳ Same letter → different position

↳ act cat



a, c, t

act ✓

tac ✓ x

act

a, c, t

atc ✓

act

6

cat dog tac god act taco, acto n strings

act ✓

HashMap

<String, ArrayList <String>>

act → cat, tac, act

dgo → dog, god

acot → taco, acto

god ?! x

dgo

taco x

acot ?? x

O(s) ↗ s is the length of each string

In order to ensure we have an anagram, sort the individual letters.

O(n) × [ ] + O(s

O/P : cat , tac , act dog , god, taco , ato

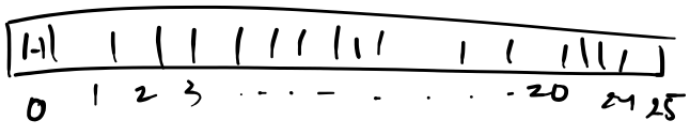
check Anagram

```
if (c1.length() != c2.length()) {
    return false;
}

int[] charCount = new int[26];

for (int i = 0; i < c1.length(); i++) {
    charCount[c1.charAt(i) - 'a']++;
    charCount[c2.charAt(i) - 'a']--;
}

for (int count : charCount) {
    if (count != 0) {
        return false;
    }
}

return true;
```

Check Anagram

→ a b c d +1

⇒ c b d a −1



actual values

act → cat , tac , act

dgo → dog , god

acOt → taco , ato

acot , act, dgo → Ans if sorted on basis

U

U of keys

X

Subarray sum divisible by K

4, 5, 0, -2, -3, 1

K = 5

5 % 5 = 0 ✓

5 % 5 = 0 ✓

5 + 0 = 5 % 5 = 0 ✓

0 % 5 = 0 ✓

0, -2, -3 = -5 % 5 = 0 ✓

-2, -3 % 5 = -5 % 5 = 0 ✓

5, 0, -2, -3 = 0 % 5 = 0 ✓

6

↳ +4 % 5

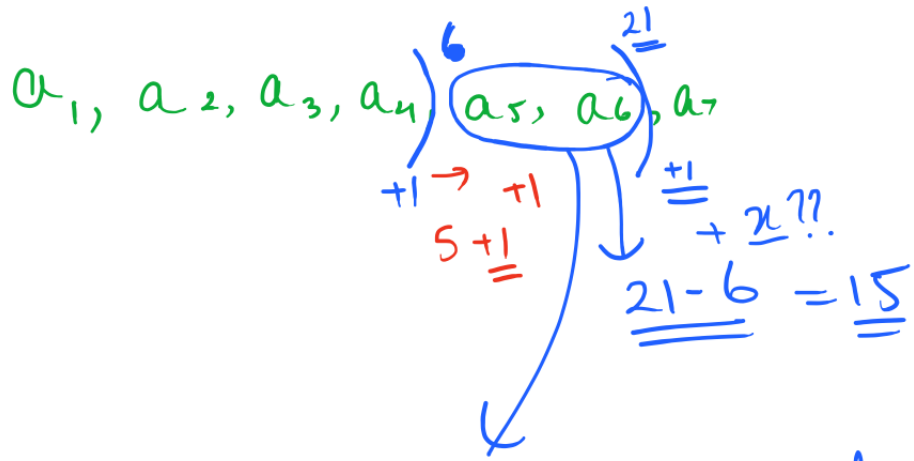
% 5

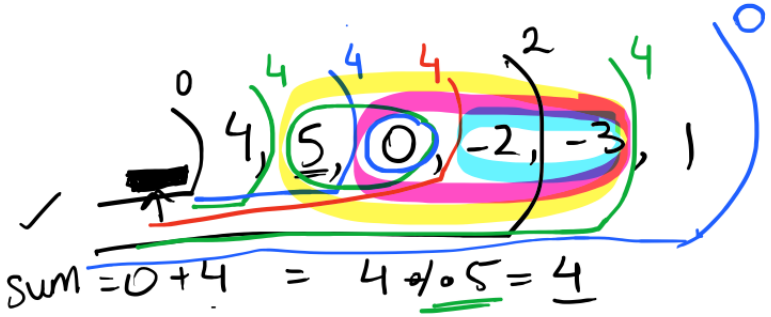
[0, 5)

$\Rightarrow 10 \,\%\, 5 = 0$

$-1 \,\%\, 5$

$\hookrightarrow (-5+4) \,\%\, 5 \Rightarrow 4 \,\%\, 5$ ✓





ans = 1+2+3+1

K = 5

Map Remainder vs Count

Sum = 4 + 5 = 9 % 5 = 4

Sum = 4 + 0 = 4

Sum = 4 - 2 = 2

Sum = 2 - 3 = 2 - 5 + 2

= 4

Sum = 4 + 1 = 5 % 5 = 0

0: ~~1~~ 2

4: ~~3~~ 4

2: 1

-2 % 5 ⇒ 5 - 3

-2 % 5 = (-2 + 5 - 5) % 5

= (-2 % 5 + k) % k

a % k