451. Sort Characters By Frequency

ex i/p → "tree"

o/p → "eetr" or "eert"

i/p → "ccc aaa"

o/p → "aaaccc" or "ccc aaa"

i/p → "Aabb"

o/p → "bbAa" or "bbaA"

Approch - 1

Do whatever question is saying.

(i) sort based on frequency

↓

first get frequency of every character.

↓

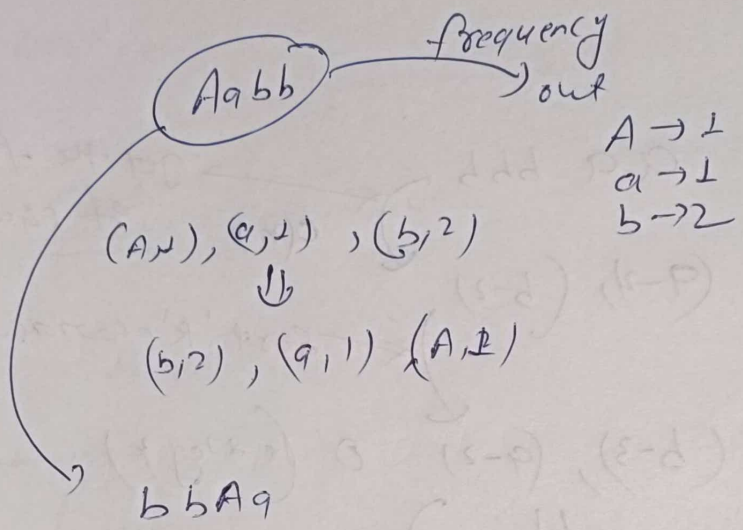
Then sort the string based on the frequency using camparator fn.

T.C. = O(n log n)

S.C. = O(log n) → while sorting heap space

Approch-2

In previous approch what we are doing is



we are sorting the entire string i.e. sorting of string length - n.

But we have only lower, upper case & digits in string i.e.

$26 + 26 + 10 \Rightarrow K$ ← we have only K distinct characters.

so rather than sorting n length string sort only K character.

how

ex → str = "aabbb" → n = 5

↓

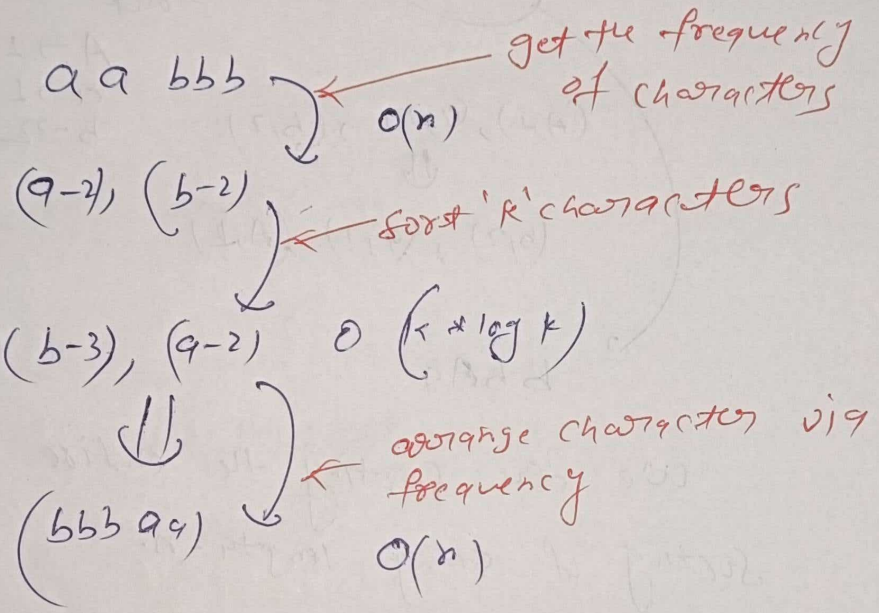
a = 2, b = 3

↳ K = 2 sort only this two characters

after sorting k disting characters,

(b-3), (a-2)

i.e.



T.C. = O(n + k log k)

S.C. = O(n)

Approach-3

So far we are optimizing the sorting it self $(n \log n) \longrightarrow (k \log k)$ & if we want more optimization then we have to optimize the sorting bcz O(N) comes

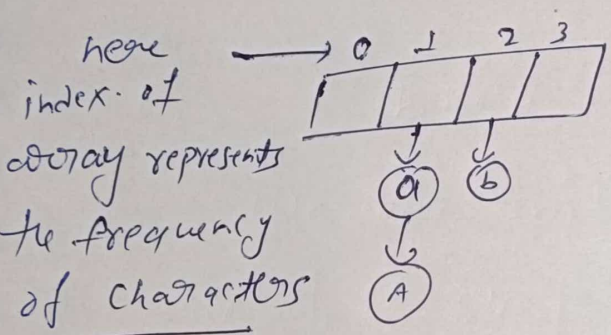
bcz we have to traver the string.

to optimizing the sorting we can use bucket sort.

Bucket Sort

use a sorting algorithm which not use the sorting time complexity.

A a b b



A character can have max freq = n

→ after this traver this array from back bcz max frequency character lies in back & append that character with their freq no. in string.

b b a A

If, problem is asked like you have to sort the same frequency character based on their ASCII value then you have to sort the every index of bucket array.

T.C. = O(n log n)

S.C. = O(n)