ALTERNATING CHARACTERS

GIVEN: Set = { ··· } ; where $\forall$ Set[element] $\in$ ascii[a-z]

PROBLEM CONSTRAINTS / FUNCTIONS CONSTRAINTS :

Set = { ··· }

↓

For table$_{ascii[a-z]}$ (Frequency Table):

- $\forall_{ascii[a-z]\ chars}$ (Frequencies are the Same) **OR**
- $\forall_{ascii[a-z]\ chars}$ (Frequencies are the Same except 1 char)
- $\forall_{frequencies}$ ($1 \leq frequency \leq 10^5$)
- **DISTINCT:** $\forall!_{frequencies\_count}$ ($1 \leq frequency\_count \leq 2$)

FIND: where the listed constraints are true and return "YES", if found, else, return "NO".

VISUALIZE PROBLEM

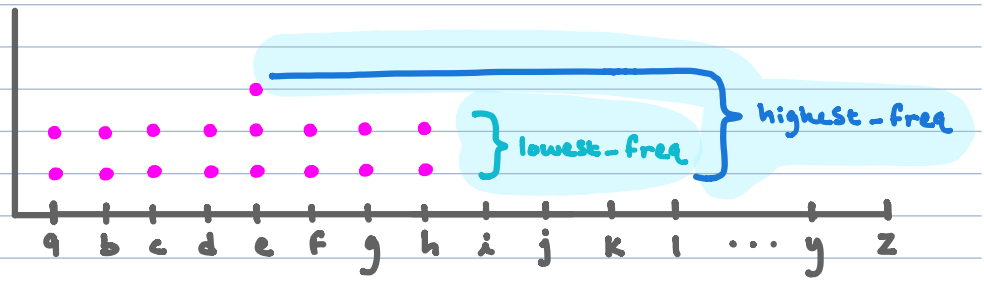
↓

MATHEMATICAL MODEL :

* (FREQUENCY PROBLEM)

Set = {a, b, c, d, e, f, g, h, h, g, f, e, d, e, c, b, a}

⇓



- @max, only 1 char's frequency differs ✓
- frequencies are in range ✓
- ! frequency_count = 2 ✓

* frequency = object's count (individually)

↳ (i.e. each char's count)

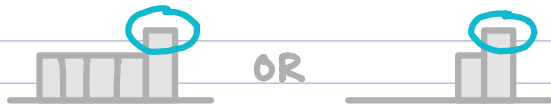
ALGORITHM: MAthematical Model Explanation

- First, create 2 empty frequency graphs (vectors) for all lowercase letters of size 26.

- Increment through the Set and store the frequency of each character into its frequency graph.
- Increment through the frequency graph and grab the highest and lowest frequencies found in it.
- Increment through the frequency graph and grab the count of times each frequency (the higher and lower) occur, respectively.
- MaKe a copy of the frequency graph and find ALL of the DISTINCT instances of frequencies

- If (highest frequency = lowest frequency) then return "YES"
- Else If (highest frequency != lowest frequency) then :
  - If the number of DISTINCT (unique) frequencies found in the frequency graph is between 1 and 2, inclusively, then:

    If case 1 : return "YES"

    high_freq = low_freq + 1
    high_freq_count = 1
    low_freq_count ≥ 1

    If case 2 : return "YES"

    low_freq = 1
    high_freq ≥ 2
    high_freq_count ≥ 1

    If case 3 : No match: return "NO"

  - Else return "NO"
- Else return "NO"