

3. feladat

A feladat címe: String reorder

A feladat linkje:

https://cses.fi/problemset/task/1743?fbclid=IwY2xjawOPyjFleHRuA2FlbQIxMABicmlkETBPaE1zMm9wS3ozRzg4aVMxc3J0YwZhcHBfaWQQMjLyMDM5MTc4ODIwMDg5MgABHjjlxAnzMBoqdH8iOsRymS9HOEn-Km4JR9hhwNV7io3RE24RnTePLIYbiC6y3_aem_aNkJFCqPxQzmw9F5LiKHJw

A feladat szövege:

Your task is to reorder the characters of a string so that no two adjacent characters are the same.
What is the lexicographically minimal such string?

Input

The only line has a string of length n consisting of characters A–Z.

Output

Print the lexicographically minimal reordered string where no two adjacent characters are the same. If it is not possible to create such a string, print -1.

A feladatod az, hogy úgy rendezd át egy karakterlánc karaktereit, hogy két szomszédos karakter ne legyen azonos. Mi a lexikografikusan legkisebb ilyen karakterlánc?

Bemenet

Az egyetlen sor egy n hosszúságú karakterláncot tartalmaz, amely az A–Z karakterekből áll.

Kimenet

Írd ki a lexikografikusan legkisebb átrendezett karakterláncot, amelyben nincs két azonos szomszédos karakter. Ha nem lehetséges ilyen karakterláncot létrehozni, írd ki: -1.

A feladat magyarázata:

Először megszámoljuk, hogy melyik betűből mennyi van a sztringben.

Ha bármelyik betű száma több, mint $(n + 1) // 2$, akkor biztosan lehetetlen olyan sztringet készíteni, ahol nincsenek egymás melletti azonos karakterek.

Ez egy ismert szükséges feltétel: ha egy betűből „túl sok” van, nem lehet őket szétosztani úgy, hogy minden között legyen más is. Ilyenkor azonnal kiírjuk: -1, és kész.

Ha elvileg lehetséges a megoldás, akkor balról jobbra felépítjük az eredmény sztringet, pozícióinként.

Minden pozíciónál a következőt csináljuk:

Megpróbáljuk betenni ide az ábécé szerint legkisebb olyan betűt, amiből még van, és nem egyezik az előzővel (hogy ne legyen két egyforma egymás mellett).

Amikor jelöltünk egy betűt, ideiglenesen kivesszük a darabszámát (csökkentjük a gyakoriságát), és megnézzük, hogy a maradék betűk között nincs-e olyan, amelyik már túl sok, tehát száma nagyobb, mint $(maradék\ hossz + 1) // 2$?

Ha még így is teljesül a „nem túl sok ugyanabból” feltétel, akkor véglegesen lerögzítjük ezt a betűt az aktuális pozícióra.

Ha nem, akkor visszavonjuk a lépést, visszaadjuk a darabszámát, és megpróbáljuk a következő betűt az ábécében.

Idő- és memóriaigény:

Minden pozíciónál legfeljebb 26 betűt próbálunk ki (A–Z), ezért a futásidő $O(26 \cdot n)$, ami gyakorlatilag $O(n)$. A memóriaigény nagyon kicsi: csak egy 26 elemű tömb, meg néhány változó. Ez bőven belefér az 1 másodperces idő- és 512 MB memórialimitbe.

A feladat megoldása:

```
import sys

s = sys.stdin.readline().strip()
n = len(s)

freq = [0] * 26
for ch in s:
    freq[ord(ch) - 65] += 1

if max(freq) > (n + 1) // 2:
    print(-1)
    sys.exit(0)

res = []
prev = -1

for pos in range(n):
    placed = False
    for c in range(26):
        if freq[c] == 0 or c == prev:
            continue
        freq[c] -= 1
        rem = n - pos - 1
        if max(freq) <= (rem + 1) // 2:
            res.append(chr(c + 65))
            prev = c
            placed = True
            break
    freq[c] += 1
    if not placed:
        print(-1)
        sys.exit(0)

print("".join(res))
```