FCHAR SOLUTION

Subtank1: Làm với cách làm "Đề bảo làm gì, ta làm nấy" ☺

Chi phí $O(n^3)$

Subtank2:

Sử dụng thuật toán với độ phức tạp $O(N^2)$. Sử dụng kĩ thuật mảng cộng dồn.

Trước tiên, khai báo mảng f[30][100007] với ý nghĩa F[c][i] là số lần xuất hiện kí tự c từ vị trí 1 đến vị trí i trong chuỗi kí tự ➔ Để tính được số lượng kí tự c từ vị trí i đến vị trí j ta có công thức O(1) như sau:

F[c][j] – F[c][i – 1] là số lượng kí tự c từ vị trí i đến vị trí j (i < j)

Ý tưởng như sau:

Với mỗi vị trí j chạy từ 1 đến n, ta tìm kiếm xem vị trí i gần nhất mà thỏa mãn được điều kiện như đề bài.

For(j = 1 → n){

For(I = j + 1 → 1){

For(c = 'a' → 'e') if(f[c][j] – f[c][I – 1] < m) break;

If(c > 'e') ans = min(ans, j – I + 1), break;

}

}

Subtank3:

Sử dụng thuật toán với độ phức tạp O(N*logN), cải tiến với ý tưởng trên bằng kĩ thuật Tìm kiếm nhị phân (Binary search algorithm).

Ta nhận thấy rằng với độ dài là L thì có đoạn kí tự (i, j) thỏa mãn được điều kiện của bài toán ➔ với độ dài L + 1 thì đoạn kí tự (I, j,) cũng hoàn toàn thỏa mãn.

Thứ ta cần tìm chính là độ dài L tìm kiếm bằng Tìm kiếm nhị phân, độ phức tạp là O(LogN), với mỗi L, ta kiểm tra xem có đoạn kí tự nào trong chuỗi kí tự với độ dài L thỏa mãn được điều kiện bài toán không. Chi phí kiểm tra là O(N)

trai = 0, phai = n + 1, L;

while(trai <= phai){

  L = (trai + phai) div 2;

 if(ok(L) = true) ans = L, phai = L – 1;

```
    else trai = L + 1;
}
```

Hàm kiểm tra ok() các bạn có thể tự viết dựa trên ý tưởng của subtank1.