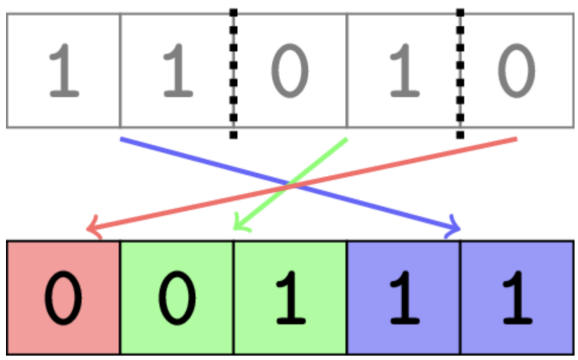
Binary Cut

June 14, 2025

1 Problem

You are given a binary string[†]. Please find the minimum number of pieces you need to cut it into, so that the resulting pieces can be rearranged into a sorted binary string.



Note that:

- each character must lie in exactly one of the pieces;
- the pieces must be contiguous substrings of the original string;
- you must use all the pieces in the rearrangement.

[†]A *binary string* is a string consisting of characters `0` and `1`. A *sorted binary string* is a binary string such that all characters `0` come before all characters `1`.

2 Solution

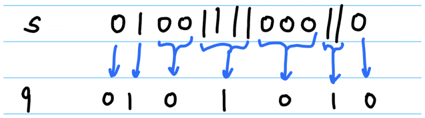
We present an $O(n)$ time solution where n is the size of input string.

```
f(s,n){
        q = s[1]
        for(i = 2;i<=n;++i){
                if(s[i] != q.back())    q.push_back(s[i]);
        }

        if(q.size >= 3){
                return q.size - 1;
        }else if(q == "10"){
                return 2;
        }else{
                return 1;
        }

}
```

By above construction, each letter of q corresponds to a consecutive substring of 0s (or 1) in s :



Notice there can never be 2 consecutive 0s (or 1s) in q.

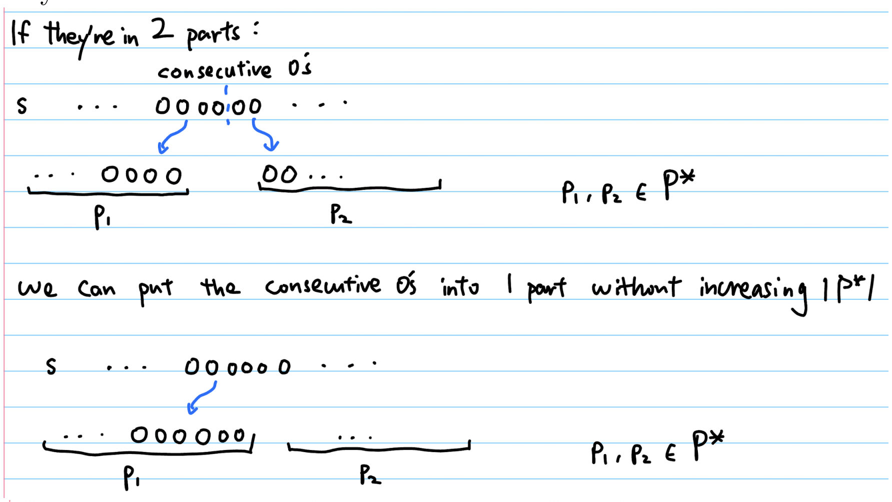
When $|q| \geq 3$, there must be a 01 substring of q, the algorithm let it be 1 piece and let all other letters in q be a piece on their own. Thus there're $|q| - 1$ pieces and clearly you can rearrange them to get sorted binary string (later we'll show this's optimal by lemma below).

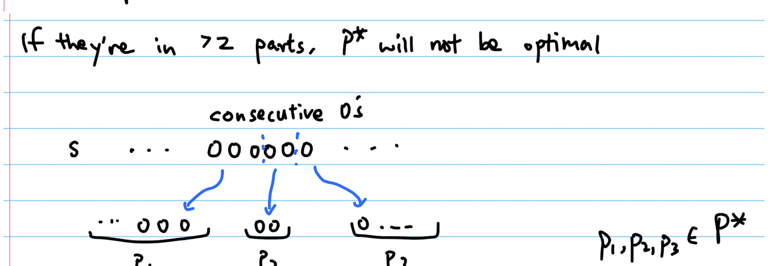
Else if $q = 10$, then s is of the form 1110000000, the optimal number of pieces is clearly 2.

Otherwise we have either $q = 01$ or $q = 0$ or $q = 1$, the optimal number of pieces is clearly 1.

Lemma. Let P^* be the pieces formed by the optimal cut. If $|q| \geq 3$, then $|P^*| \geq |q| - 1$.

proof. Suppose $|q| \geq 3$ and $|P^*| < |q| - 1$. Now for every letter in q, which corresponds to a consecutive substring of 0s (or 1) in s, WLOG we can assume all of them belongs to exactly 1 part in P^*, we used picture to illustrate why :





and clearly they cannot be in 0 part. In other words, every letter of q is contained in exactly 1 piece in P^*.

It follows either (i) there's a part $p \in P^*$ that contains > 2 letters of q or (ii) there're > 1 part of P^* than contain 2 letters of q. (Otherwise, all parts of P^* contain ≤ 1 letter of q except possibly 1 part that may contain 2 letters of q, this would means $|P^*| \geq q - 1$).

For (i), it's easy to see such part must contain 3 consecutive letters of q. However, any 3 consecutive letters of q is either of the form 101 or 010, clearly they cannot be in the same part.

For (ii), suppose they're p_1, p_2, it's easy to see p_1 must contain 2 consecutive letters of q (same for p_2). However, any 2 consecutive letters of q must be either of the form 10 or 01, so there're no way to sort p_1, p_2 a contradiction.
■