You are given a binary string(*i.e.* with characters `0` and `1`) S consisting of characters $S_1, S_2, \ldots, S_N$. In a single operation, you can choose two indices L and R such that $1 \le L \le R \le N$ and flip the characters $S_L, S_{L+1}, \ldots, S_R$. By flipping, we mean change character `0` to `1` and vice-versa.

You aim is to perform ATMOST one operation such that in final string number of `1` s is maximised. If you don't want to perform the operation, return an empty array. Else, return an array consisting of two elements denoting L and R. If there are multiple solutions, return the lexicographically smallest pair of L and R.

Notes:
- Pair (a, b) is lexicographically smaller than pair (c, d) if a < c or, if a == b, then b < d.