

761. Special Binary String

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

- *Difficulty: Hard*
- Total Accepted: 585
- Total Submissions: 1.8K
- Contributor: [cnmnwpu](#)
- [Subscribe](#) to see which companies asked this question.

Related Topics : String, Brute Force

Special binary strings are binary strings with the following two properties:

- The number of 0's is equal to the number of 1's.
- Every prefix of the binary string has at least as many 1's as 0's.

Given a special string *S*, a *move* consists of choosing two consecutive, non-empty, special substrings of *S*, and swapping them. (*Two strings are consecutive if the last character of the first string is exactly one index before the first character of the second string.*)

At the end of any number of moves, what is the lexicographically largest resulting string possible?

Example 1:

Input: *S* = "11011000"

Output: "11100100"

Explanation:

The strings "10" [occurring at *S*[1]] and "1100" [at *S*[3]] are swapped. This is the lexicographically largest string possible after some number of swaps.

Note:

1. *S* has length at most 50.
2. *S* is guaranteed to be a *special* binary string as defined above.

Use brute force to find two consecutive special strings *s1* and *s2*. Swap them if the latter string *s2* is lexicographically bigger than the former string *s1*.

This might work, have fun :)

```
string makeLargestSpecial(string S) {  
    string ans = S;  
    string curr = S;  
    while(makeSpecial(curr,ans)){
```

```

    curr = ans;

}

return ans;

}

```

```

bool makespecial(string s, string &result)
{

```

```

    //                S = [s0] [  s1  ] [  s2   ] [s3]
    // string index :          i0    i1    j0      j1
    //
    // swap s1 and s2 if s2>s1. After swap store it in result, return true

```

```

for(int i0=0;i0<(int)s.size();++i0)
{
    int n0=0; int n1 = 0;
    for(int i1=i0;i1<(int)s.size();++i1)
    {
        if(s[i1]=='1') n1++;
        else n0++;
        if(n1<n0) break;
        else if (n1==n0)
        {
            string s1 = s.substr(i0,i1-i0+1);
            int j0=i1+1;
            int m0=0;
            int m1=0;
            for(int j1=j0;j1<(int)s.size();j1++)
            {
                if(s[j1]=='1') m1++;
                else m0++;
                if(m1<m0) break;
                else if(m1==m0)
                {

```

```

        string s2 = s.substr(j0,j1-j0+1);
        if(s2>s1)
        {
            string s0 = s.substr(0,i0);
            string s3 = s.substr(j1+1);
            result = s0+s2+s1+s3;
            return true;
        }else{
            break;
        }
    }
}
}
}
}
return false;
}

```