

Flip Game

You are playing the following Flip Game with your friend: Given a string that contains only these two characters: `+` and `-`, you and your friend take turns to flip two **consecutive** `++` into `--`. The game ends when a person can no longer make a move and therefore the other person will be the winner.

Write a function to compute all possible states of the string after one valid move.

For example, given `s = "++++"`, after one move, it may become one of the following states:

```
[  
  "--++",  
  "+--+",  
  "++--"  
]
```

If there is no valid move, return an empty list `[]`.

Solution 1

```
public List<String> generatePossibleNextMoves(String s) {  
    List list = new ArrayList();  
    for (int i=-1; (i = s.indexOf("++", i+1)) >= 0; )  
        list.add(s.substring(0, i) + "--" + s.substring(i+2));  
    return list;  
}
```

written by [StefanPochmann](#) original link [here](#)

Solution 2

We start from the second character of the input string and check whether current and previous characters are both equal to '+'. If true, then we combine the characters : characters before previous character + '--' + characters after current character.

```
public List<String> generatePossibleNextMoves(String s) {  
    List<String> list = new ArrayList<String>();  
    for (int i = 1; i < s.length(); i++) {  
        if (s.charAt(i) == '+' && s.charAt(i - 1) == '+') {  
            list.add(s.substring(0, i - 1) + "--" + s.substring(i + 1, s.length()  
));  
        }  
    }  
    return list;  
}
```

written by [lDreaml](#) original link [here](#)

Solution 3

```
def generatePossibleNextMoves(self, s):  
    return [s[:i] + "--" + s[i + 2:] for i in xrange(len(s) - 1) if s[i:i + 2] ==  
            '++']
```

25 / 25 test cases passed.

Status: Accepted

Runtime: 44 ms

It is a simple list comprehension and a filter

written by [dietpepsi](#) original link [here](#)

From [LeetCoder](#).