214. Shortest Palindrome

W.l.o.g, let's assume string s is $\overline{a_0a_1...a_n...a_1a_0bcde...} = AA^{-1}B$. Our goal is finding longest palindrome AA^{-1}, which starts at a_0. And final answer is $B^{-1}AA^{-1}B$.

There are 2 solutions, 1st is a simple brute force method, 2nd is a recursive method which can match s and its reverse much faster. It uses one loop($O(N)$) to find longest match C and the other part D of s, AA^{-1} must be prefix of C , and D must be prefix of B , then call the function recursively, $D^{-1} + f(C) + D$, to get right answer.

1st solution:

```
public String shortestPalindrome(String s) {
      String r = new StringBuilder(s).reverse().toString();
      int loc = 0;
      for(; loc < r.length()-1; loc++)
          if(r.substring(loc, r.length()-1).compareTo(s.substring(0,
           s.length()-1-loc)) == 0)
              break;
      return r.substring(0, loc).concat(s);
}
```

2nd solution:

```
public String shortestPalindrome(String s) {
        //String r = new StringBuilder(s).reverse().toString();
        int loc = 0;
        for(int i = s.length()-1; i >= 0; i--){
          if(s.charAt(loc) == s.charAt(i)){
              loc++;
          }
        }
        if(loc == s.length()){
            return s;
        }
        return new StringBuilder(s.substring(loc,s.length())).reverse
        ().toString() + shortestPalindrome(s.substring(0,loc))
         + s.substring(loc,s.length());
}
```