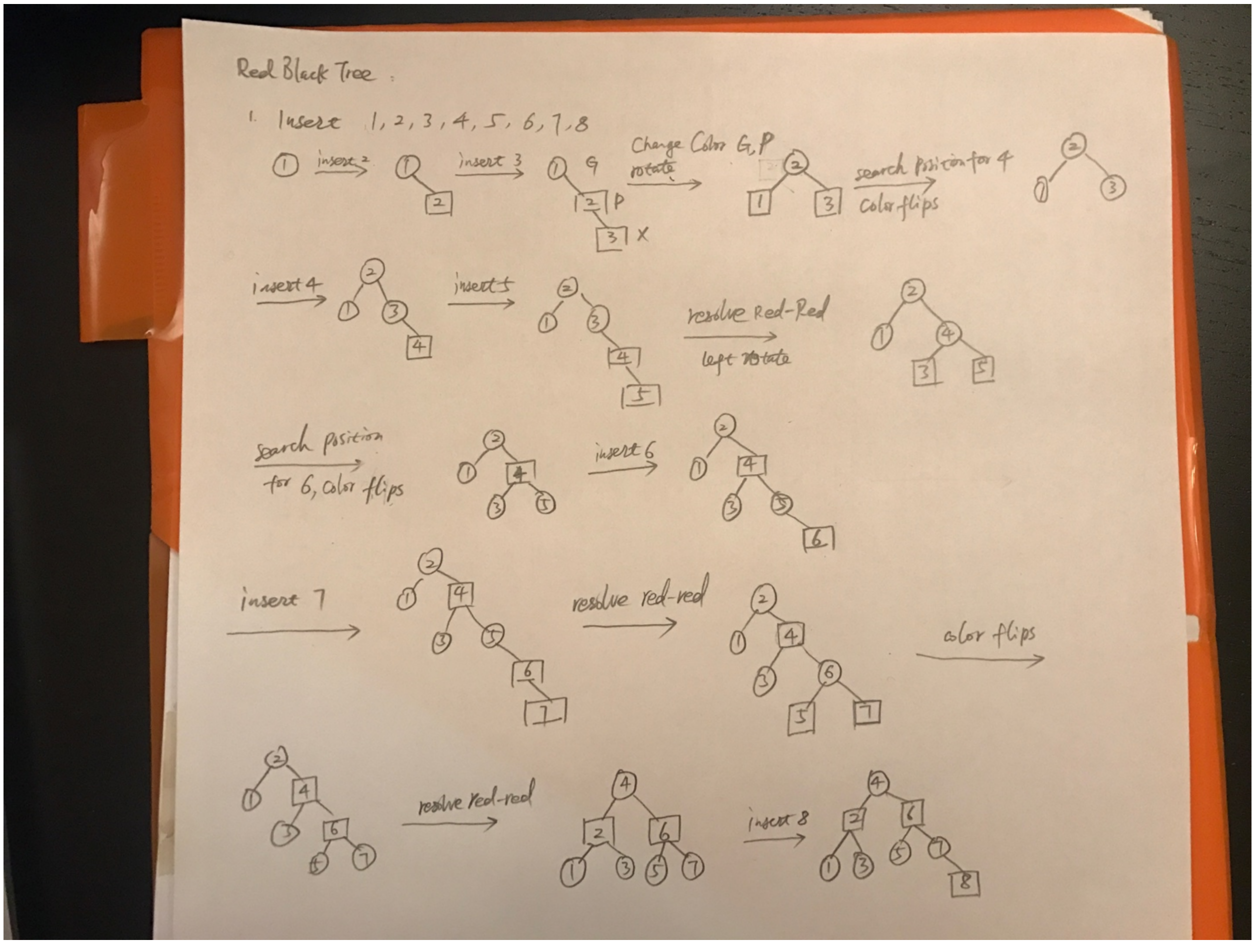
Yuliang Jin 986381

Problem 1

[Interview Question] Devise an O(n) algorithm to accomplish this task: Given a none-empty string S of length n, S consists some words separated by spaces. We want to reverse every word in S. For example, given S = “we test coders”, your algorithm is going to return a string with every word in S reversed and separated by spaces. So the result for the above example would be “ew tset sredoc”.

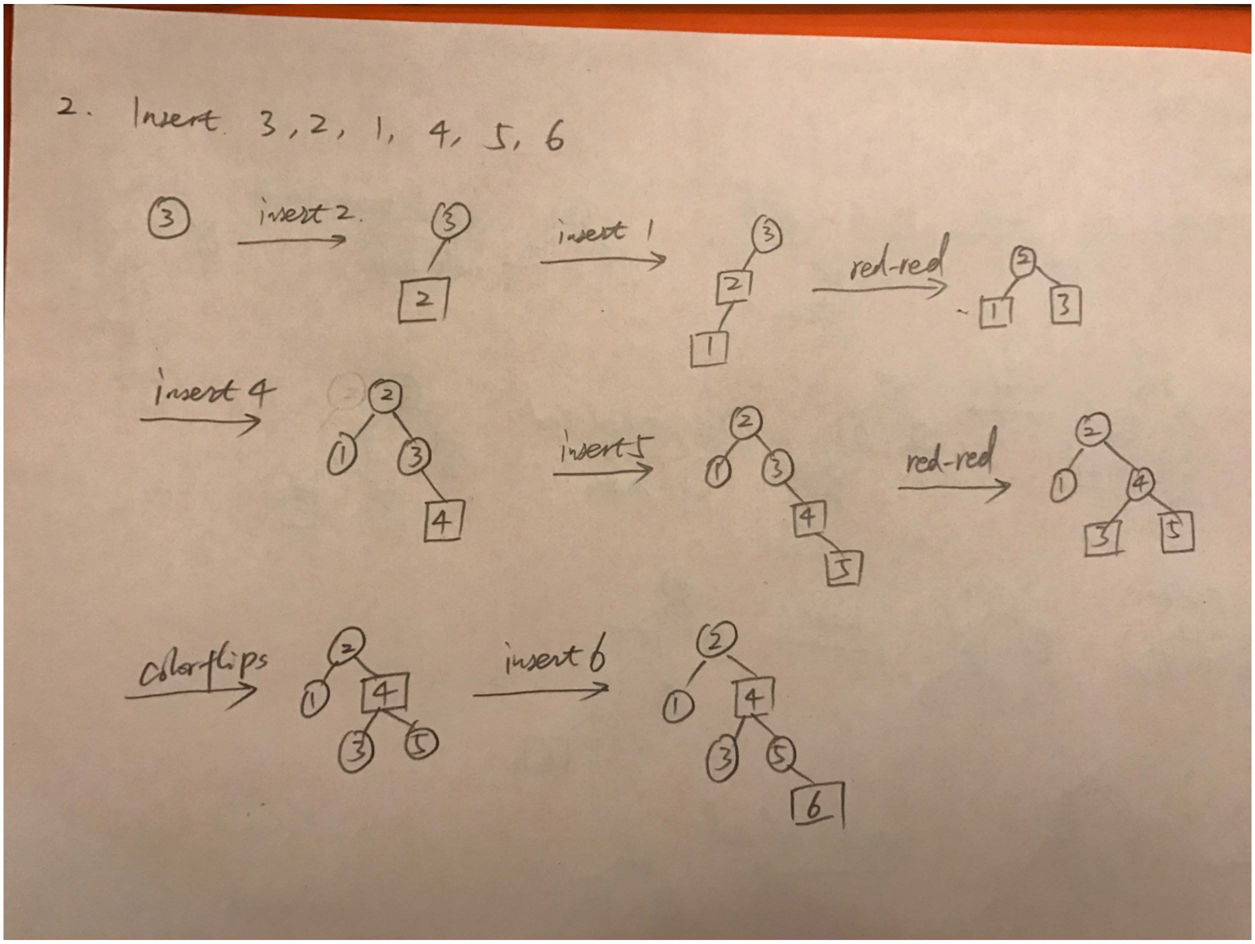
```
public static String reverseByStack(String s) {
    StringBuilder stringBuilder = new StringBuilder();
    Stack<Character> characters = new Stack<Character>();
    char[] chars = s.toCharArray();
    for(int i=0;i<chars.length;i++) {
      if(' ' != chars[i]) {
        characters.push(chars[i]);
      } else {
        while (!characters.isEmpty()) {
          stringBuilder.append(characters.pop());
        }
        stringBuilder.append(' ');
      }
    }

    while (!characters.isEmpty()) {
      stringBuilder.append(characters.pop());
    }
    return stringBuilder.toString();
  }
```

Problem 2



Red Black Tree :

1. Insert 1, 2, 3, 4, 5, 6, 7, 8



Problem 3

```
public static boolean isPrime(int number) {
    int sqrt = (int) Math.sqrt(number) + 1;
    for (int i = 2; i < sqrt; i++) {
      if (number % i == 0) {
        return false;
      }
    }
    return true;
  }
```

Time time complexity is O(n ^ (1/2)).

Problem 4

A. IsPrime(n) is O(L) in terms of input size.

B.