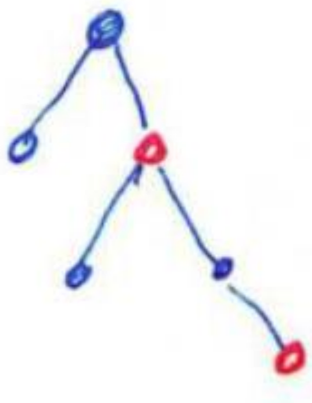
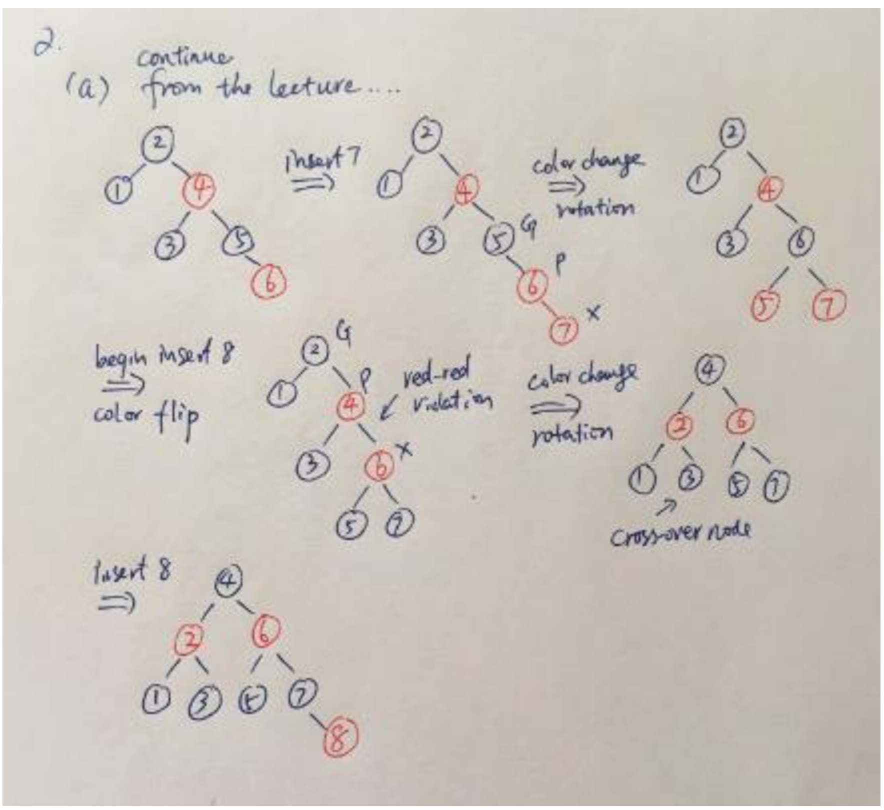
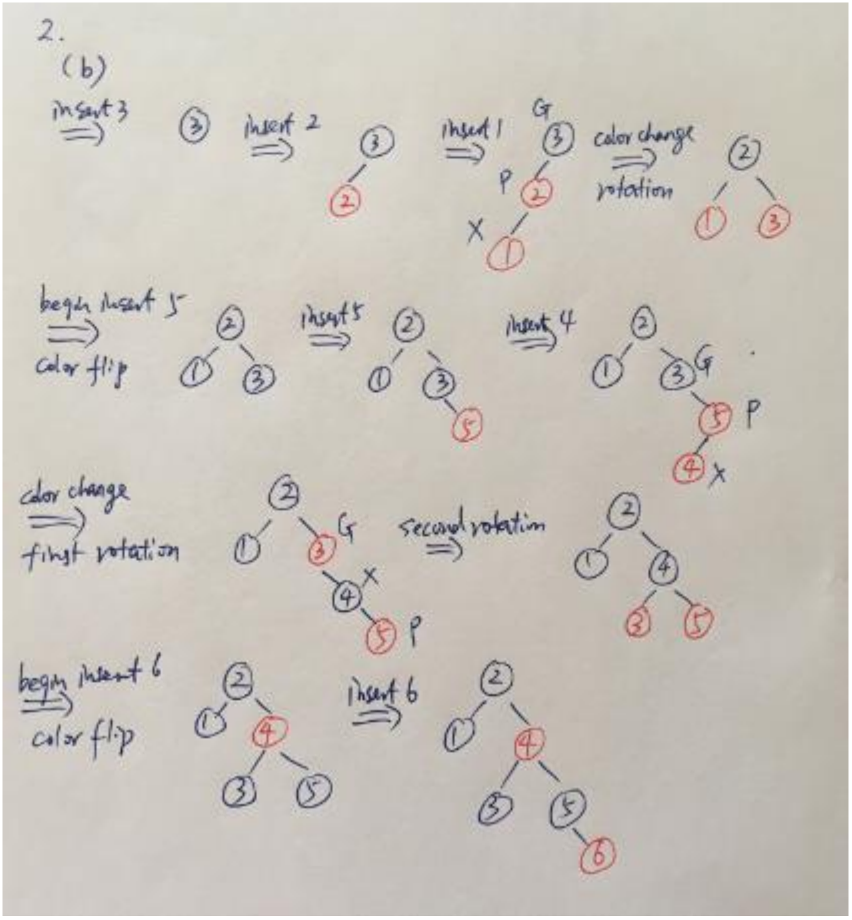
Lab 8 Solutions

Problem 1.



Problem 2.





Problem 3.

```
//Precondition: n is a positive integer
public static boolean isPrime(int n) {
    if(n == 2) return true;
    if(n== 1 || n % 2 == 0) return false;
    //check the odd numbers <= sqrt(n)
    for(int i = 3; i * i <= n ; i = i+2) {
        if(n % i == 0) return false;
    }
    return true;
}
```

Problem 4.

(A) Express the asymptotic running time of your algorithm `IsPrime(n)` in terms of the input size rather than input value.

Solution. The running time in terms of input value is given by $T(n) = n^{\frac{1}{2}}$. Since the number b of bits in a positive integer n is $\Theta(\log n)$, it follows that n is $\Theta(2^b)$ and so running time in terms of b is given by

$$T(b) = \Theta(\left(2^b\right)^{\frac{1}{2}}) = \Theta(2^{\frac{b}{2}}).$$

(B) Suppose $T(b)$ is the running time of your algorithm in terms of input size. Show that b^2 is $o(T(b))$. (It can be shown that b^k is $o(T(b))$ for any positive integer k. Consequently, this algorithm is said to run in *superpolynomial* time.)

Solution. Using L'Hopital twice we have:

$$\begin{aligned}
\lim_{n\to\infty} \frac{b^2}{2^{\frac{b}{2}}} &= \lim_{n\to\infty} \frac{2b}{\frac{1}{2}\cdot 2^{\frac{b}{2}}\cdot \ln 2} \\
&= \lim_{n\to\infty} \frac{2}{\frac{1}{4}\cdot 2^{\frac{b}{2}}\cdot \ln^2 2} \\
&= 0.
\end{aligned}$$