Weekly Programming Challenge 1

Next Palindrome

Let us first define what we mean by a palindrome.

Definition : An integer is a palindrome if it has the same value even after reversing all its digits.
Examples :
14541, 33, 9870789 are palindromes but 23, 933, 89989 are not palindromes.

Design a JAVA program which, for a positive integer n, computes the smallest integer greater than n which is a palindrome. Your program should run efficiently for arbitrarily large value of `n`. More specifically, declare n as variable of type long, for any arbitrarily large value of n, your program should take at most a few seconds (in fact millisecond if possible).

For example :
for $n = 1234501234$, the output should be 1234554321. for $n = 325986$, the output should be 326623.
A naive approach to solve the problem would be the following :

```
check if n+1 is a palindrome
if not, check if n+2 is a palindrome
if not, check if n+3 is a palindrome
.
.
and so on.
```

The java program based on the above approach will have a `while` loop which will terminate when we find a palindrome. In each iteration we increment n and check if n is a palindrome. If yes, we terminate the loop. We may use some Boolean variable which is initialized as false, and set to true when we find a palindrome. The condition of `while` loop will be `(flag==false)`. The exact JAVA code of this program is also available at the course website (Lecture 7), and was discussed and executed in the class itself.

It can be easily seen that this program will eventually terminate since there are infinite palindromes. The correctness of the program is thus obvious. But the time taken may be huge for some inputs. Compile and execute the program on some *bad* inputs to convince yourself. The reason for its slowness is that the number of iterations of the `while` loop in this program may be very large. For example, for $n = 123456789000000000$, the next palindrome will be 123456789987654321, and in this case the number of iterations will be 987654321. So this program won't give you answer in seconds. It may even take hours/days for such inputs. So we realize that the above program/algorithm is very inefficient.

So the challenge is to design an efficient program/algorithm to find the next palindrome for a given integer n.