public static int reverse(String a, String b)

{

int la=a.length();

int lb=b.length();

if(la!=lb)

{

return 0;

}

else

{

for(int i=0; i<la; i++)

{

if(!a.substring(i,i+1).equals(b.substring(la-1-i, la-i)))

{

return 0;

}

}

}

return 1;

}

The run time of exercise one is O(n). I used one for loop in this algorithm, which has a time complexity of O(n) because the maximum number of primitive operations depending on the variables is n. I tested the length of the two strings first, and return zero immediately since if their length is not the same, they can never be reversed. I used a for loop with time complexity of O(n) to iterate through the entire string and returns zero once the string is not reversed. If zero is not reversed, one will be returned meaning that it passed all cases. Other parts of my algorithm have a time complexity of O(1), therefore the time complexity of the entire algorithm is O(n).