15,15

1. digit_list = sorted(digit_list)
n = len(digit_list)
mid = n // 2
if n % 2 == 0:

15, 1 "n % 2 == 0"
c[1]
Score: 0.05

0, 14 "(n % 2 != 0)"
c[2]
Score: 0.97

15,1

2. left_side = digit_list[:mid]
left_side_reverse = left_side[::-1]
candidate = left_side + left_side_reverse

0,14

4. return digit_list

15, 1
Score: 0.05

15,1

3. if candidate > digit_list:

15, 1 "(candidate <= digit_list)"
Score: 0.05

15,1

7. for i in range(mid - 1, -1, -1):

15, 1 "range(mid - 1, -1, -1)"
c[4]
Score: 0.05

15,1

9. if digit_list[i] < 9:

15, 1 "digit_list[i] < 9"
c[5]
Score: 0.05

15,1

11. digit_list[i] += 1
if i != n - i - 1:

15, 1 "i != n - i - 1"
c[6]
Score: 0.05

15,1

13. digit_list[n - i - 1] = digit_list[i]

15, 1
Score: 0.05

15,1

14. return digit_list[:mid] + [digit_list[mid]] * (n % 2) + digit_list[mid - 1::-1]

next_palindrome