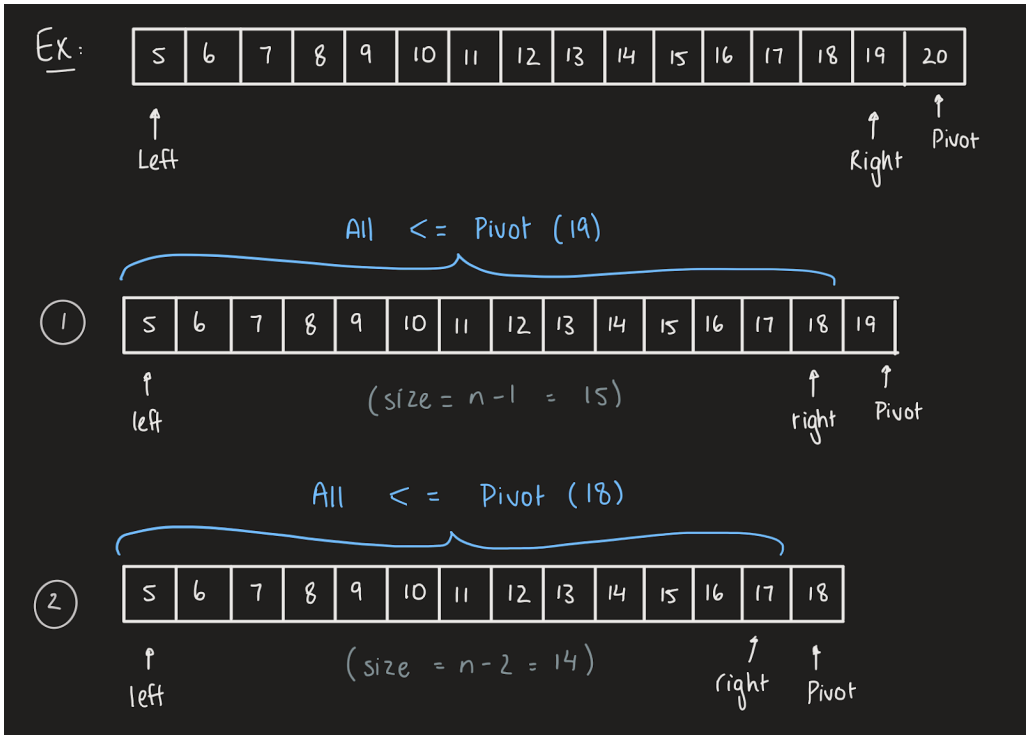
1. Worst case complexity for a sorting algorithm using quick sort occurs when the smallest or largest element in the array is chosen as the pivot each time. This causes the sub arrays to constantly consist of *n* - 1 and 1 element(s) respectively. Knowing this, we can derive the time complexity:

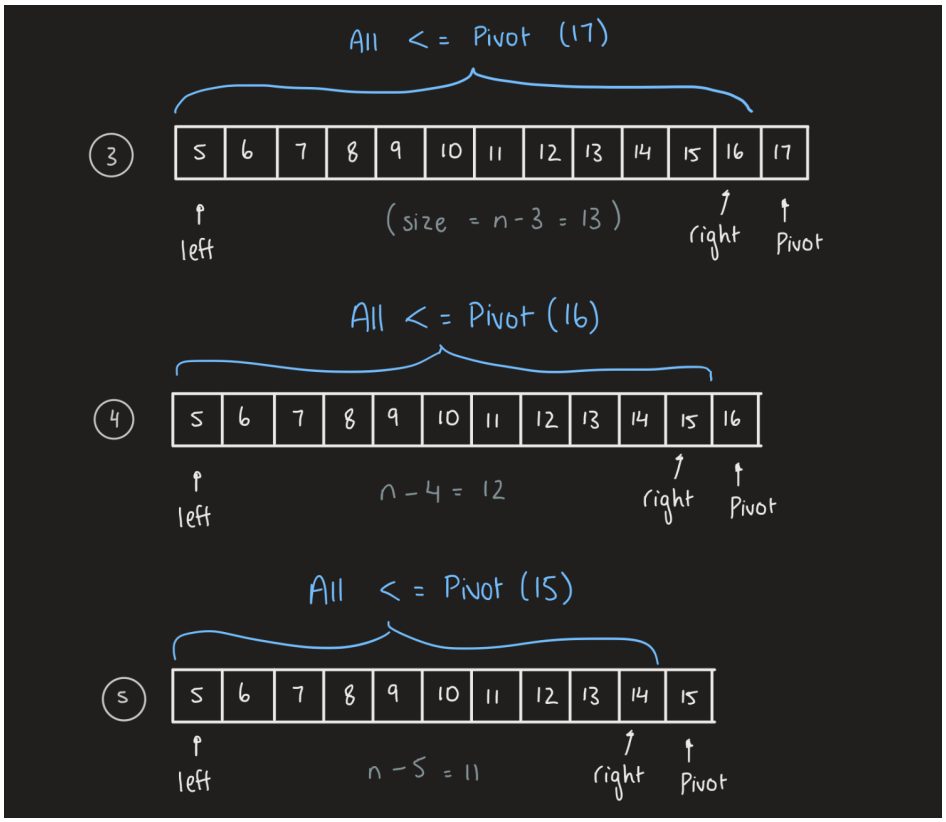
   $n + (n - 1) + (n - 2) + (n - 3) + \ldots + 2$

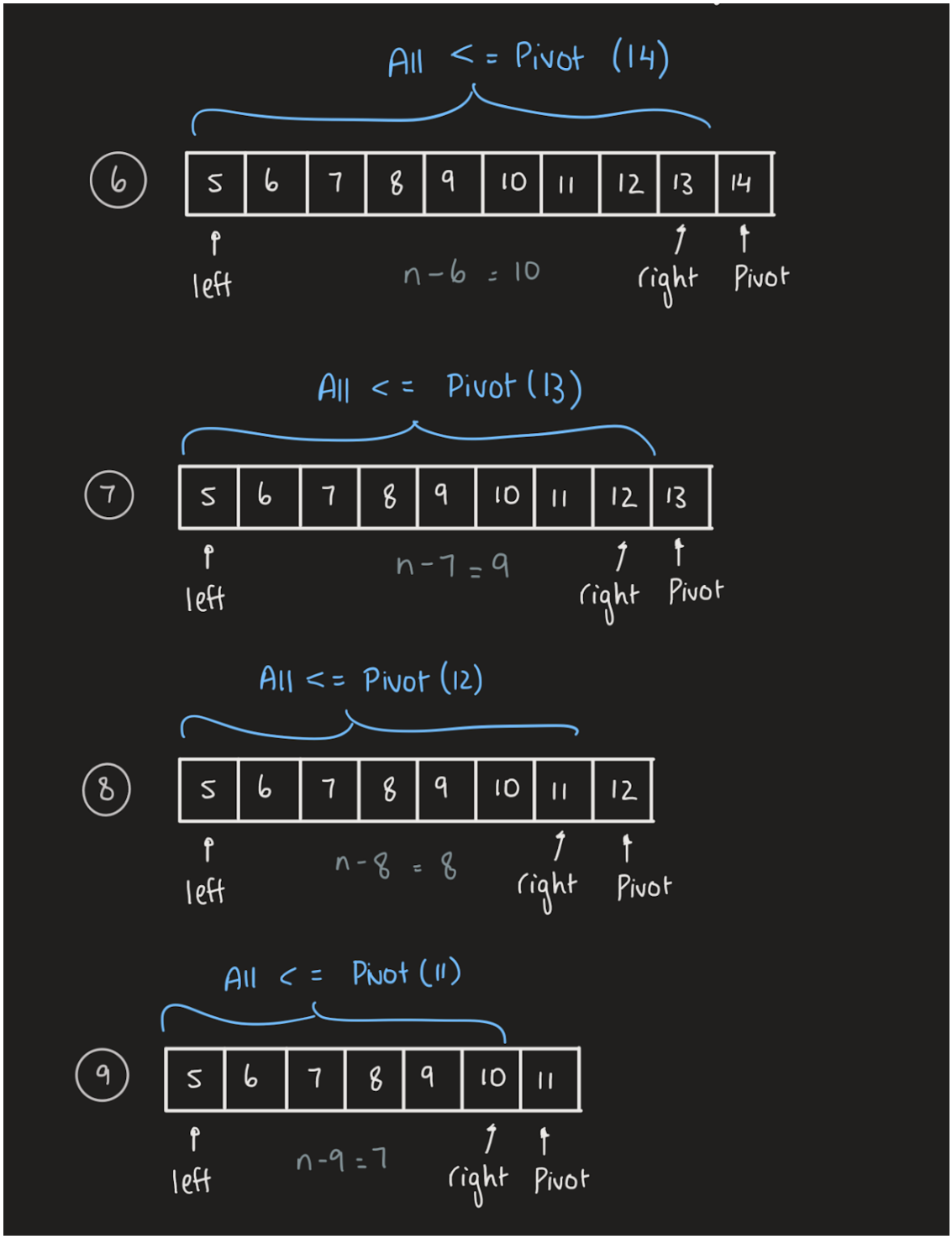
   $= \frac{n(n+1)}{2} - 1$

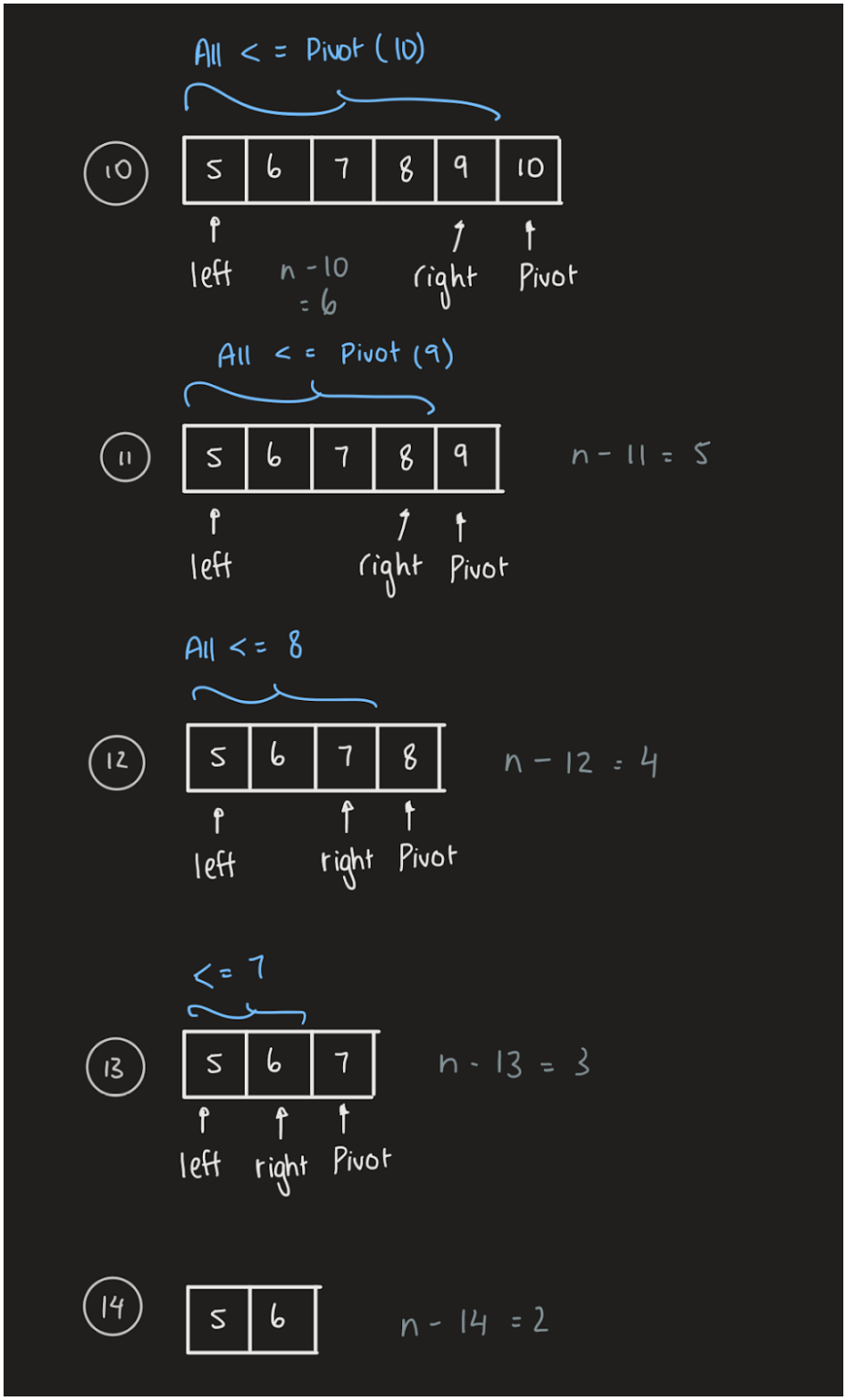
   $= O(n^2)$

2.

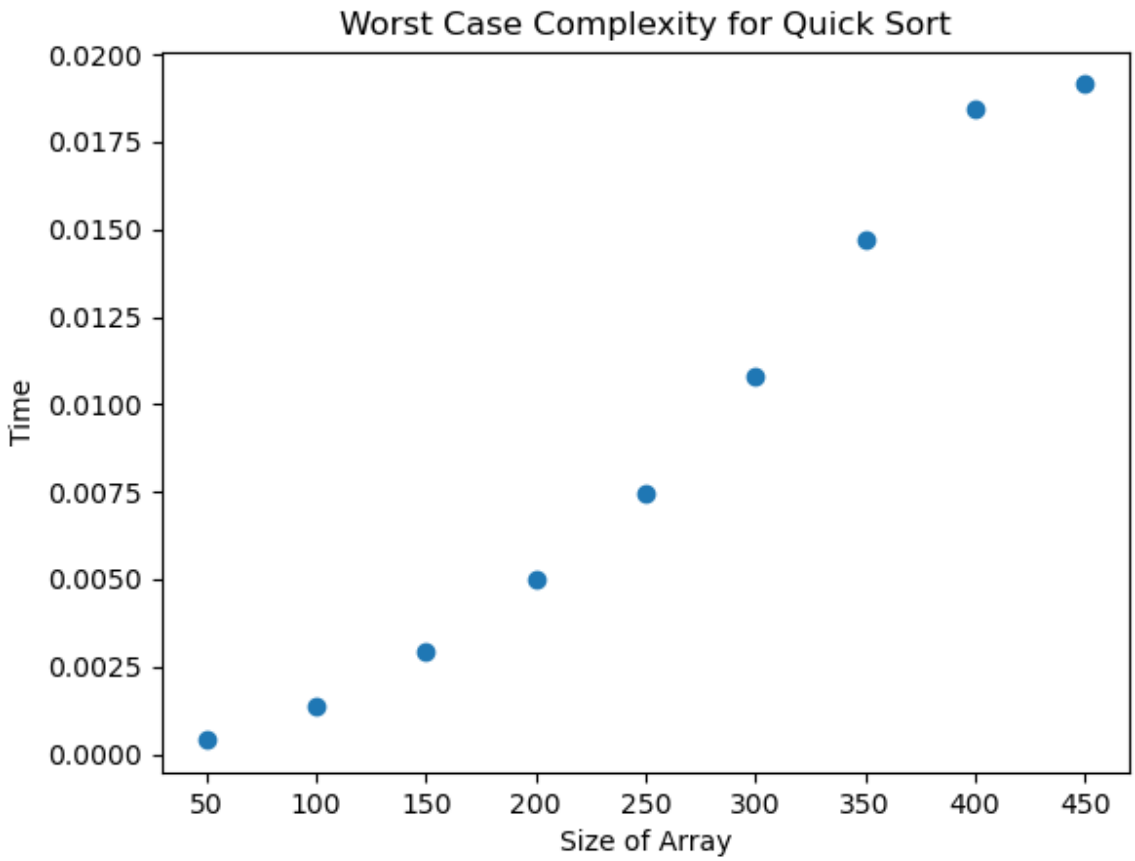

Ex:
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Left
Right
Pivot
All <= Pivot (19)
1
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
left
(size = n-1 = 15)
right
Pivot
All <= Pivot (18)
2
5 6 7 8 9 10 11 12 13 14 15 16 17 18
left
(size = n-2 = 14)
right
Pivot


All <= Pivot (17)
3
5 6 7 8 9 10 11 12 13 14 15 16 17
left
(size = n-3 = 13)
right
Pivot
All <= Pivot (16)
4
5 6 7 8 9 10 11 12 13 14 15 16
left
n-4 = 12
right
Pivot
All <= Pivot (15)
5
5 6 7 8 9 10 11 12 13 14 15
left
n-5 = 11
right
Pivot


All <= Pivot (14)
6
5 6 7 8 9 10 11 12 13 14
left
n-6 = 10
right
Pivot
All <= Pivot (13)
7
5 6 7 8 9 10 11 12 13
left
n-7 = 9
right
Pivot
All <= Pivot (12)
8
5 6 7 8 9 10 11 12
left
n-8 = 8
right
Pivot
All <= Pivot (11)
9
5 6 7 8 9 10 11
left
n-9 = 7
right
Pivot


All <= Pivot (10)
10
5
6
7
8
9
10
left
n - 10
= 6
right
Pivot
All <= Pivot (9)
11
5
6
7
8
9
n - 11 = 5
left
right
Pivot
All <= 8
12
5
6
7
8
n - 12 = 4
left
right
Pivot
<= 7
13
5
6
7
n - 13 = 3
left
right
Pivot
14
5
6
n - 14 = 2

4.



→ The graph is approximately quadratic, matching the worst case complexity for quicksort, which is $O(n^2)$.