Question 1

Given the worst case, every partition would result in one sub-array containing one element, and another sub-array containing the remaining elements. The algorithm will continue partitioning until left with two elements where it no longer will need to traverse the sub-array. This results in a formula of the form:

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

which simplifies to

$$[n * (n + 1)] / 2 - 1$$

and is simply $O(n^2)$.

Question 2

An example of the worst case would be an already sorted list and an implementation of quick sort that takes the pivot from the lower index.

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

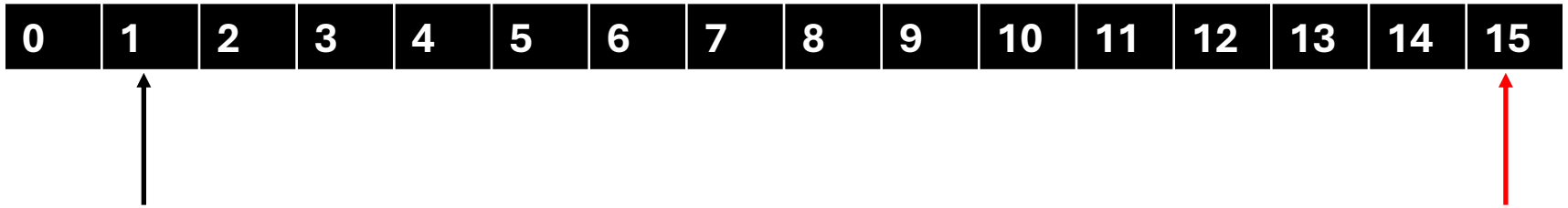
Question 2

Question 2

Left ptr

Right ptr

Pivot: 0

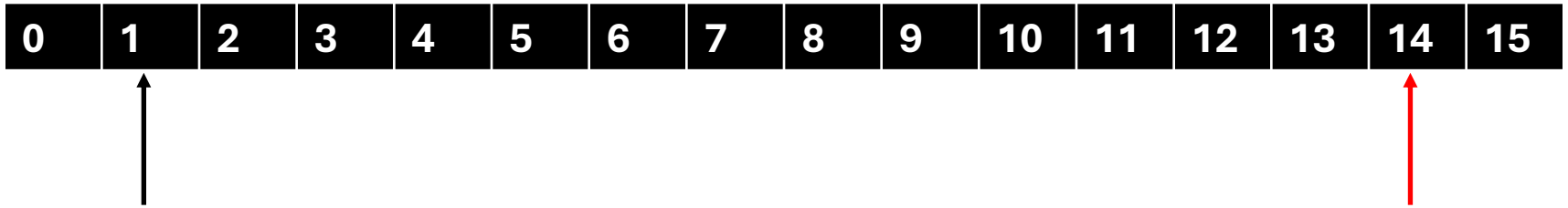


Question 2

Left ptr

Right ptr

Pivot: 0

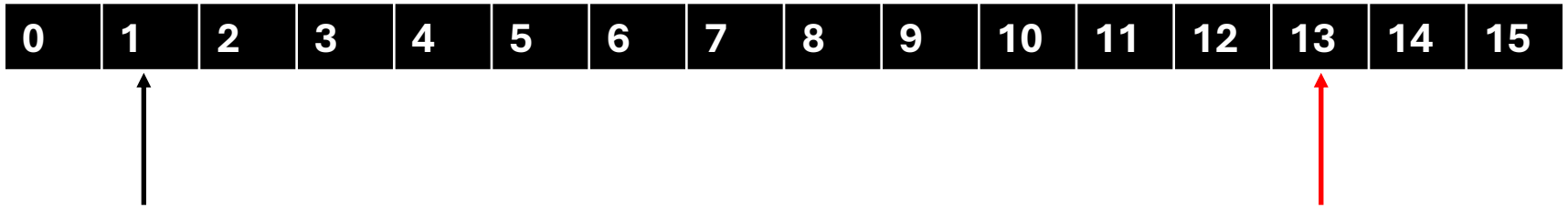


Question 2

Left ptr

Right ptr

Pivot: 0

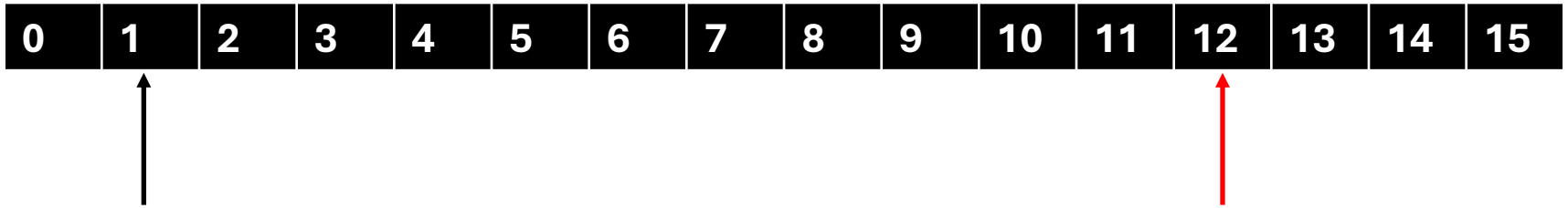


Question 2

Left ptr

Right ptr

Pivot: 0

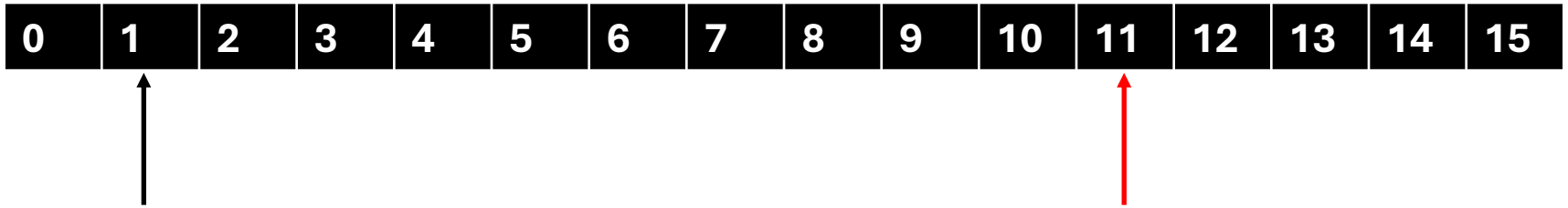


Question 2

Left ptr

Right ptr

Pivot: 0

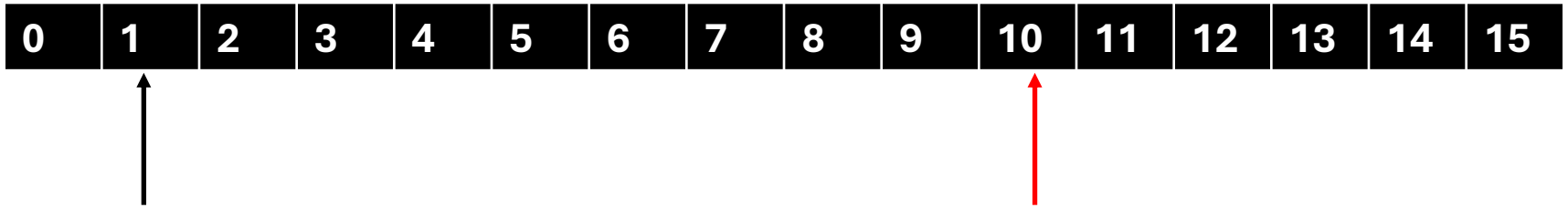


Question 2

Left ptr

Right ptr

Pivot: 0

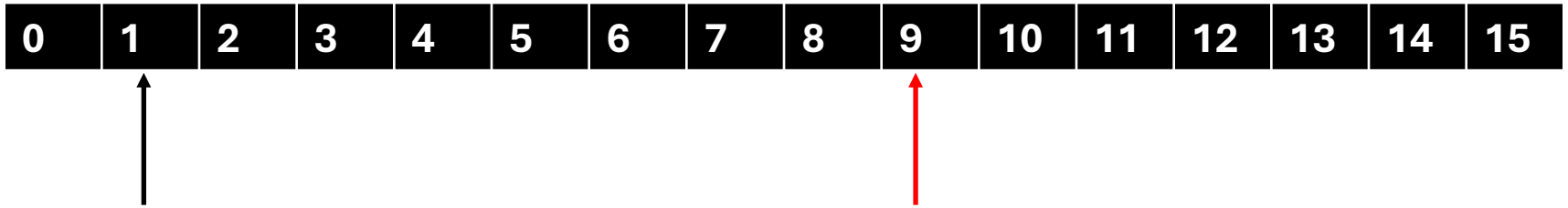


Question 2

Left ptr

Right ptr

Pivot: 0

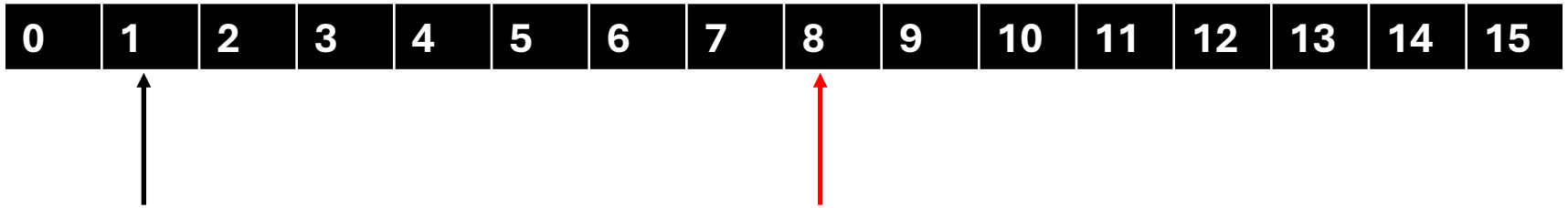


Question 2

Left ptr

Right ptr

Pivot: 0

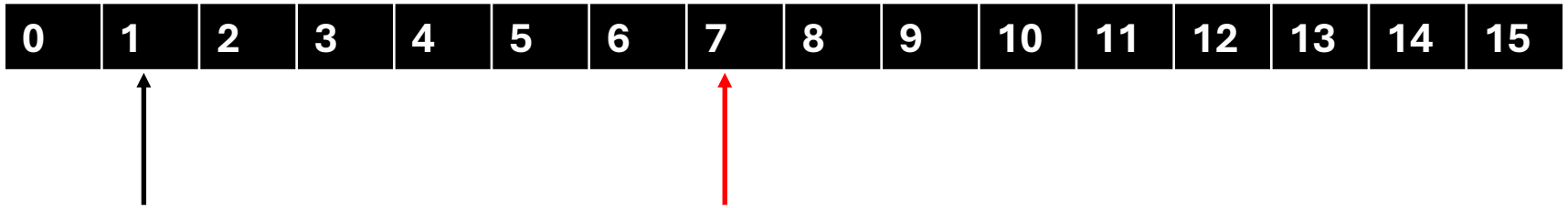


Question 2

Left ptr

Right ptr

Pivot: 0

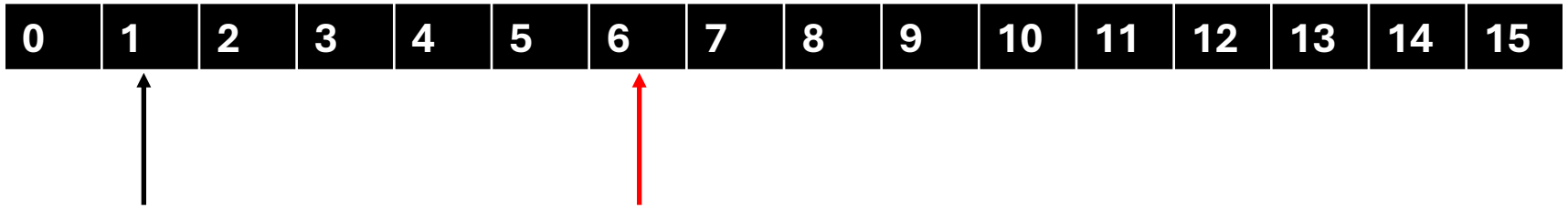


Question 2

Left ptr

Right ptr

Pivot: 0

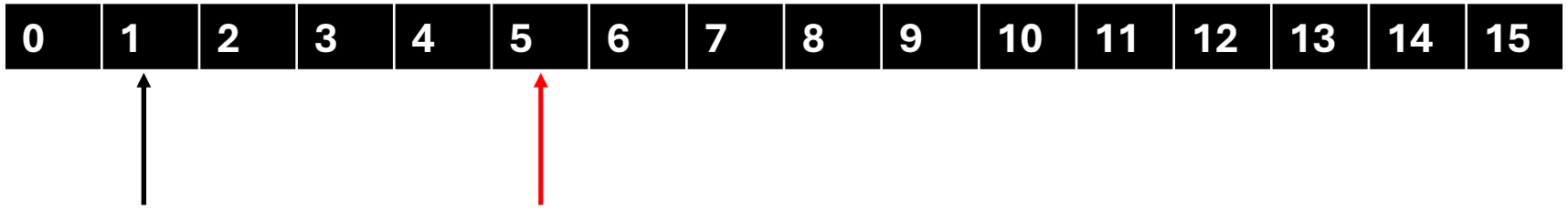


Question 2

Left ptr

Right ptr

Pivot: 0

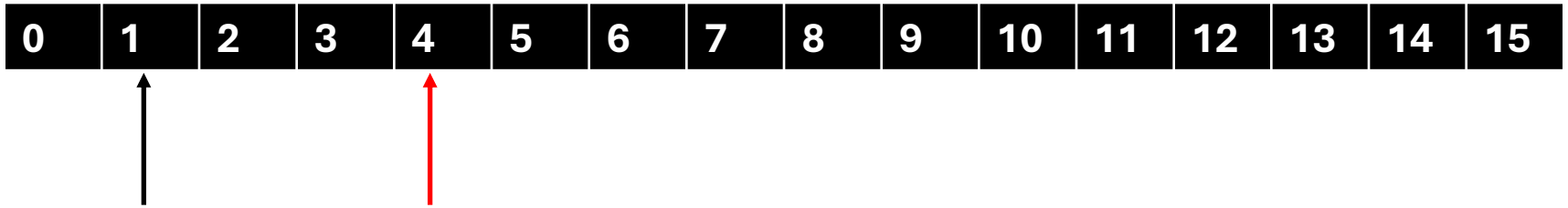


Question 2

Left ptr

Right ptr

Pivot: 0

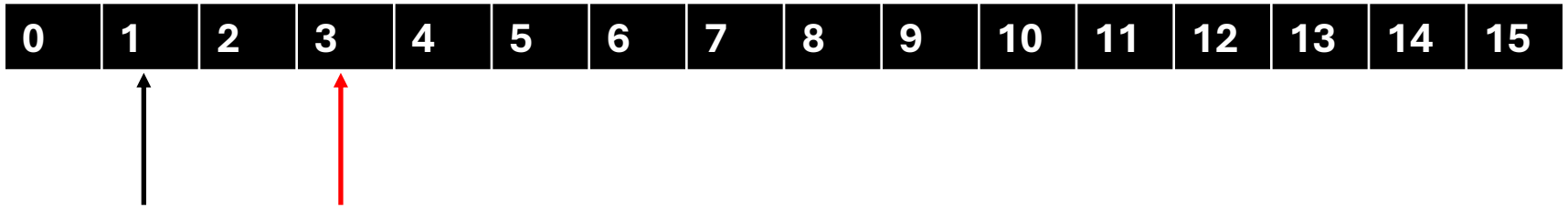


Question 2

Left ptr

Right ptr

Pivot: 0

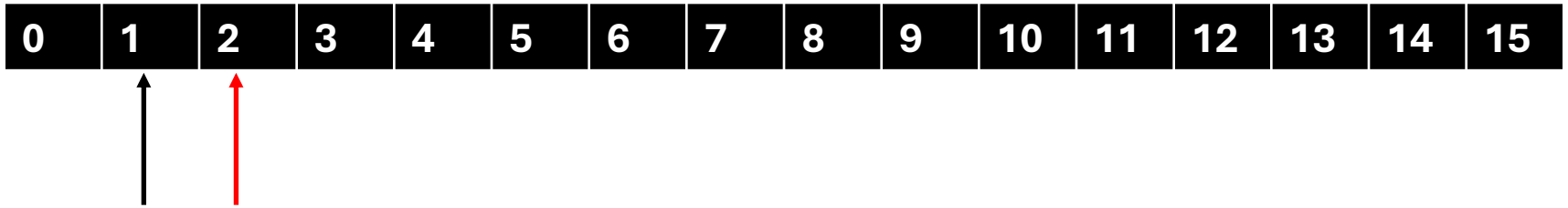


Question 2

Left ptr

Right ptr

Pivot: 0

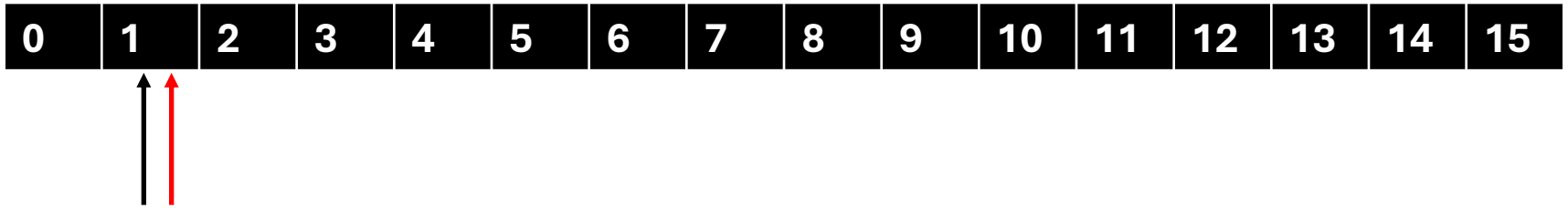


Question 2

Left ptr

Right ptr

Pivot: 0

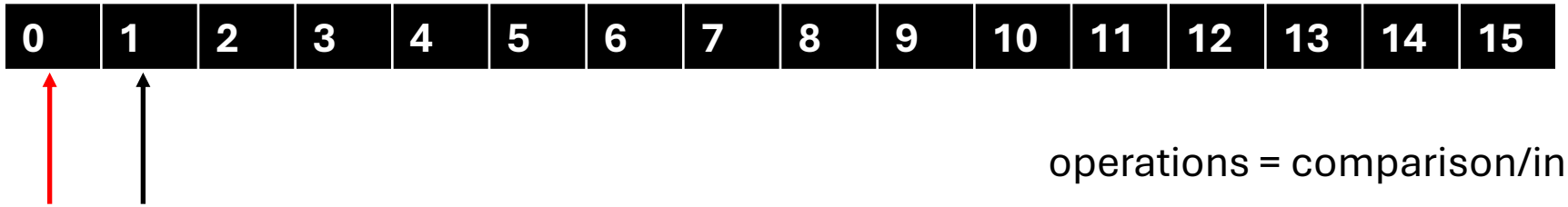


Question 2

Left ptr

Right ptr

Pivot: 0



N operations

operations = comparison/insertion/swap

Question 2

Left ptr

Right ptr

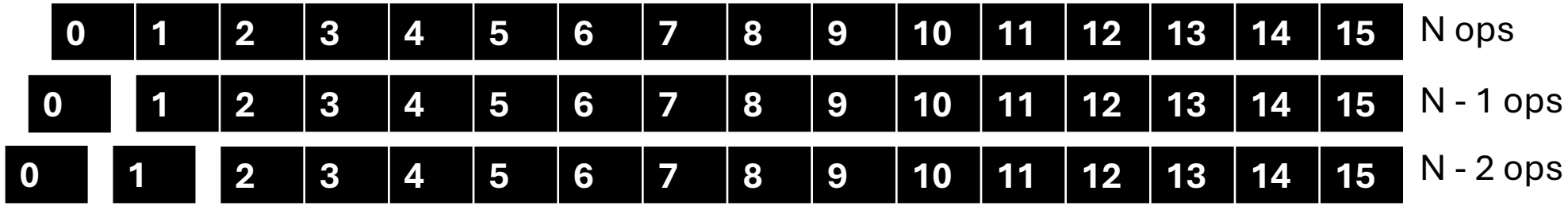
Pivot: 1

Question 2

Left ptr

Right ptr

Pivot: 1

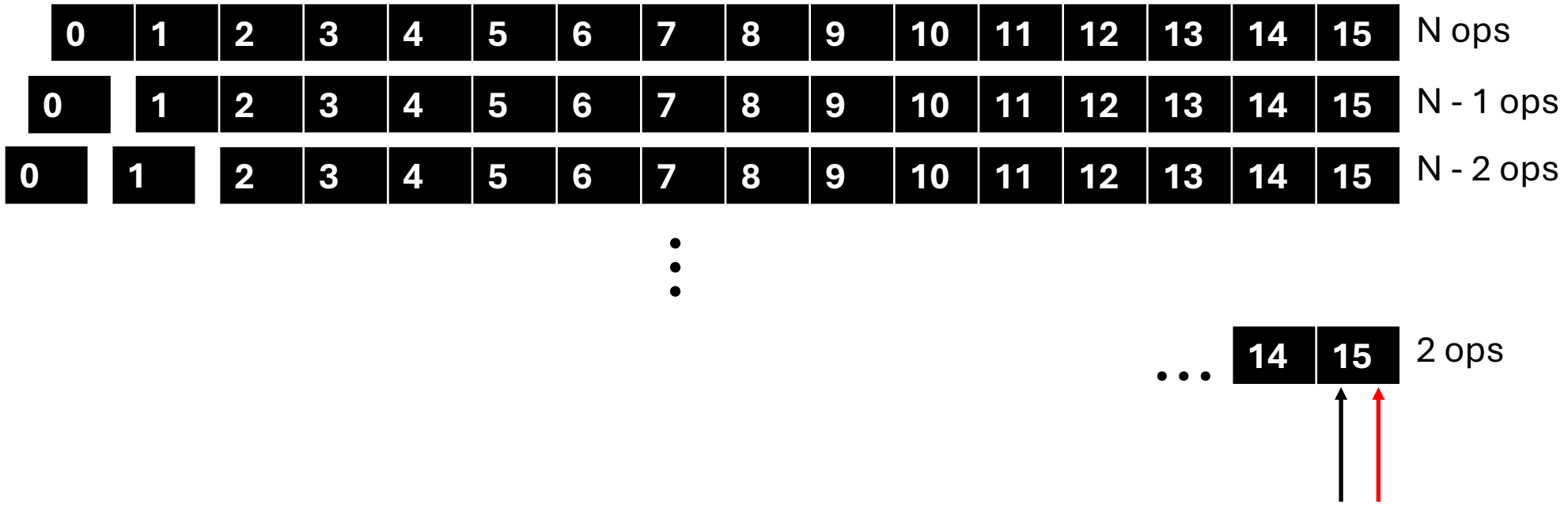


Question 2

Left ptr

Right ptr

Pivot: 1



Question 4

The plot displays a quadratic function, inline with what we expect of $O(n^2)$.