0	1	2	3	4	5	6
6	3	9	5	2	8	7

i, pivotIndex   ↑ pivot

dacă   arr[i] <= pivot

     swap (arr[i], arr[pivotIndex])

      pivotIndex ++

  i++

0	1	2	3	4	5	6
6	3	9	5	2	8	7

  i, pi]

0	1	2	3	4	5	6
6	3	9	5	2	8	7

     i, pIndex

9 > 7   =)

0	1	2	3	4	5	6
6	3	9	5	2	8	7

       pIndex i

5 < 7 =)   swap (arr[2], arr[3])

0	1	2	3	4	5	6
6	3	5	9	2	8	7

       pIndex i

2 < 7 =)   swap (arr[3], arr[4])

	0	1	2	3	4	5	6
	6	3	5	2	9	8	7

↑Index i

8 > 7 =>

	0	1	2	3	4	5	6
	6	3	5	2	9	8	7

pivotIndex i

7 <= 7 =>

	0	1	2	3	4	5	6
	6	3	5	2	7	8	9

↑Index i

return $pIndex - 1$

QuickSort (arr, st, dr)

   daca st < dr :
      pIndex = partition (arr, st, dr)
      QuickSort (arr, st, pIndex -1)
      QuickSort (arr, pIndex +1, dr)