Revision:

How is ~ 5 calculated?

$$Soln: \sim (101)_2 = (010)_2$$
$$= (2)_{10}$$

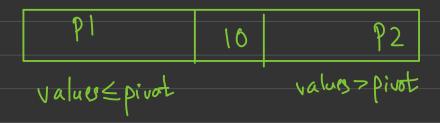$$\begin{array}{c} \sim \\ 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{array}$$

https://www.programiz.com/java-programming/bitwise-operators#:~:text=Java%20Bitwise%20Complement%20Operator,0%20and%200%2 0to%201.&text=It%20is%20important%20to%20note,%2D%20(N%20%2B%201).

$$\sim (5) = -6$$

$$5 = 0101 \quad (\text{in binary})$$

$$\sim 5 = \sim (0101)$$

$$\boxed{\sim 5 = 1010} \quad \text{i.e. } 10$$

Now we will prove that $10 \approx -6$

$$6 = 0110 \quad (\text{in binary})$$

1's C of $6 = 1001$

$$2's C = 1's C + 1 = \begin{array}{r} 1 \\ 1001 \\ + 0001 \\ \hline 1010 \end{array}$$

Quick Sort / Partition Hoare Sort.

A

10	15	1	2	9	16	11

P1	10	P2

values ≤ pivot   values > pivot

---

pivot = 7

	A	B	C	D	E	F	G	H	I	J
	lb								ub	
1	0	1	2	3	4	5	6	7	8	
2	7	6	10	5	9	2	1	15	7	swap(10,7)
3	s	s	s						e	
4	7	6	7	5	9	2	1	15	10	swap(9,1)
5			s	s	s		e	e	e	
6	7	6	7	5	1	2	9	15	10	swap(7,2)
7				s	s e	s s				
8	2	6	7	5	1	7	9	15	10	return 5
9					e	s				

values ≤ pivot   Pivot   values > pivot

partition (A, lb, ub)
{ pivot = A[lb];
  start = lb;
  end = ub;
  while(start < end)
  { while (a[start] ≤ pivot)
        start++
    while (a[end] > pivot)
        end --
    if (start < end)
        swap (a[start], a[end]);
  }
  swap (a[lb], a[end]);
  return end ;
}

---

quicksort (A, lb, ob)
{ if (lb < ob)
    loc = partition(A, lb, ob);
    quicksort (A, lb, loc-1);
    quicksort (A, loc+1, ob);
}