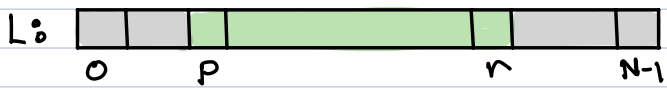
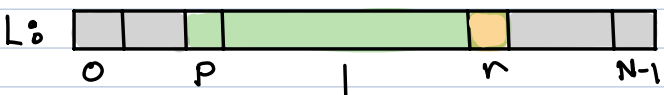
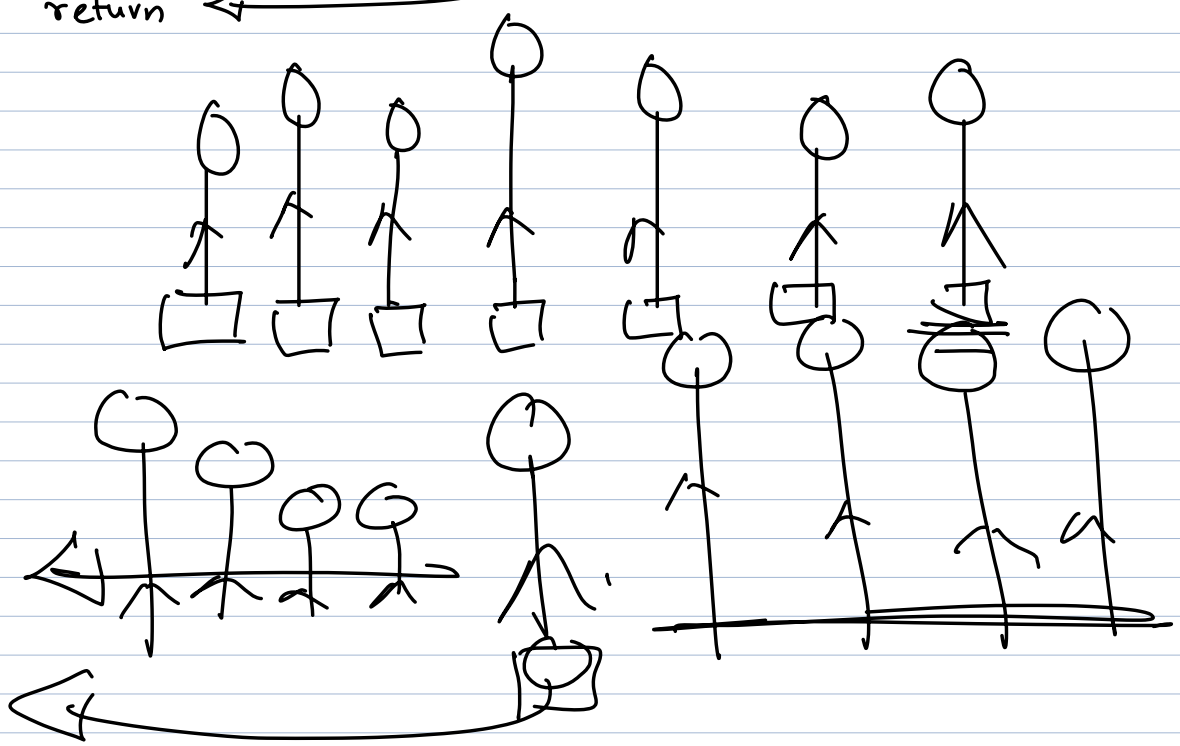


Precondition : $0 \leq p < r < N$



Re-arrange

return

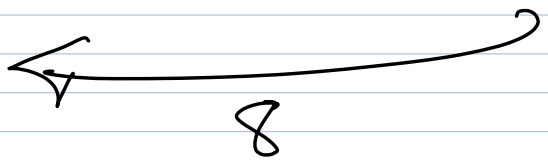


p=

4	5	6	7	8	9	10	11 (r=)
5ft	5.2ft	4.8ft	6ft	4.3ft	5.1ft	8.2ft	5.0ft

Partition

4	5	6	7	8	9	10	11
5ft	4.8	4.3	3.2	5ft	5.2	6ft	5.1



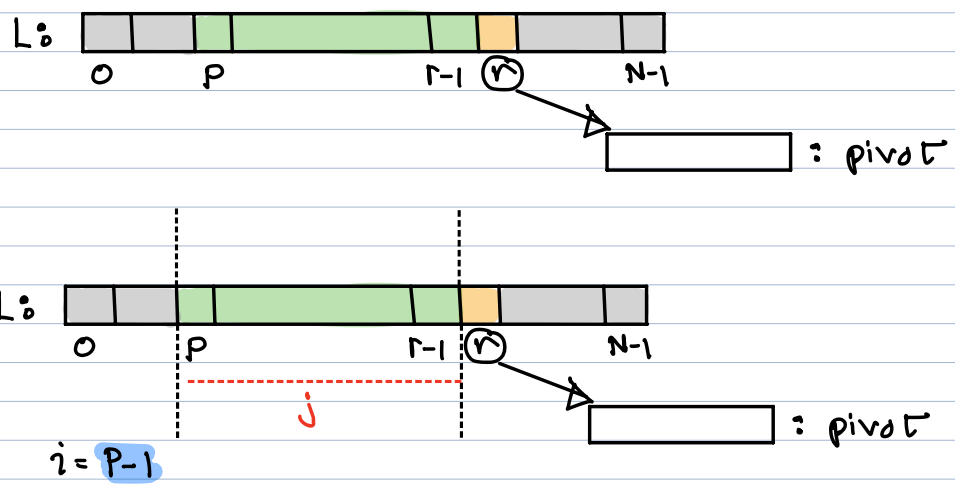
8

Tony Hoare : Computer Scientist : Quick Sort.

1970's: Communicating Seq. Processes (CSP) : Go. goroutine / async def / Co-routine

---

Adversary Input. Randomized Quick.

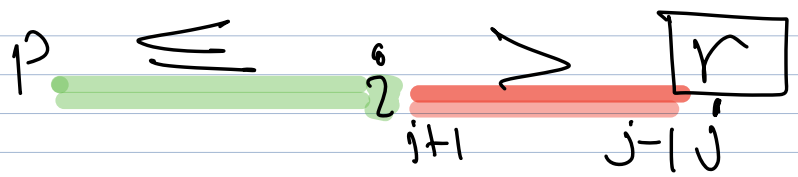


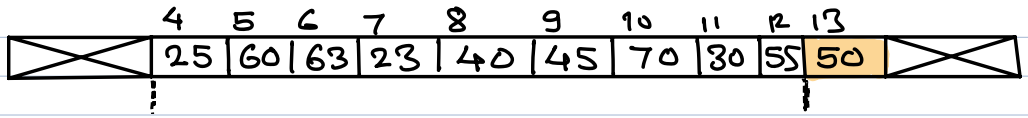
L[i] ↔ L[j]

```
j = p
while j < r :
    L[j]
    j = j+1
```

```
j = p
while j < r :
    if L[j] <= pivot:
        j = j+1
```

1) Set index $j \leftarrow p$.
   Set index $i \leftarrow p-1$
   Define tmp variable for swapping
   Set pivot $\leftarrow L[r]$
2) Visit all elements from p to r-1 using index j.
   2-1) Compare $L[j] <= pivot$.
   2-2) Condition false
   ↳ Do nothing
   ↳ Goto 2-4)
   2-3) Condition True.
   $i \leftarrow i+1$
   $L[i] \leftrightarrow L[j]$
   Goto 2-4)
   2-4) Go to the next element.

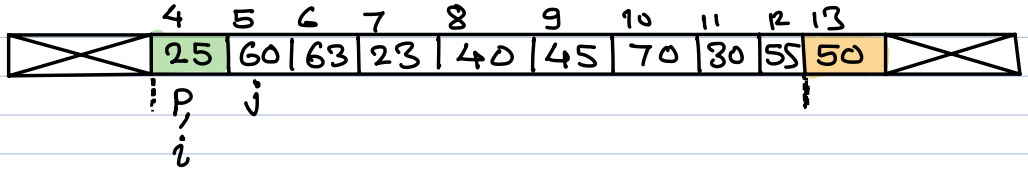




j = 4, i = 3, pivot = 50 : L[j] <= pivot : 25 <= 50 : True.

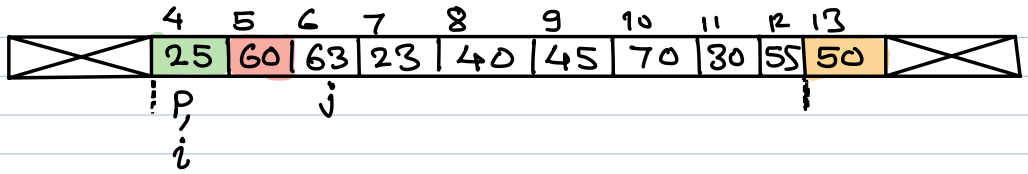
i = i+1 → i = 4, L[j] ↔ L[i]

L[4] ↔ L[4]



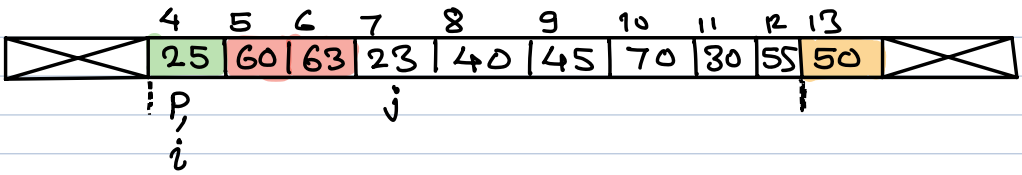
j = 5, i = 4, pivot = 50 : L[j] <= pivot : 60 <= 50 : False.

j ← j+1, Goto next



j = 6, i = 4, pivot = 50 : L[j] <= pivot : 63 <= 50 : False

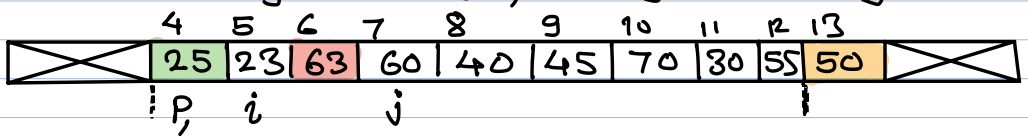
j ← j+1, Goto next



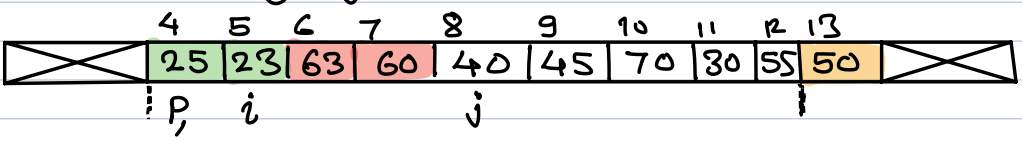
j = 7, i = 4, pivot = 50, L[j] <= pivot : 23 <= 50 : True.

i ← i+1, i == 5

L[j] ↔ L[i], L[5] ↔ L[7]



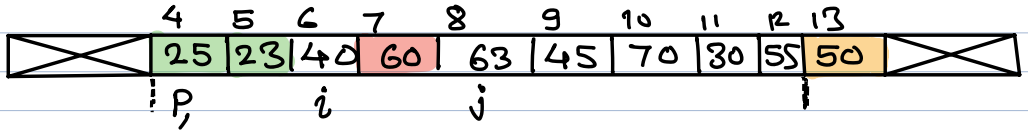
j = j+1



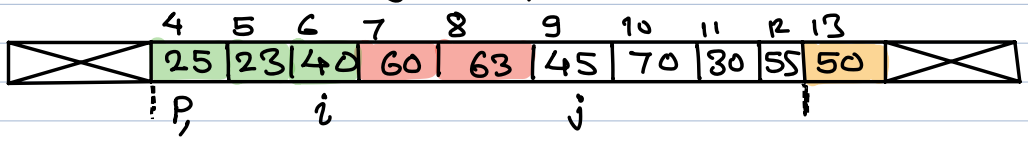
j = 8, i = 5, pivot = 50, L[j] <= pivot, 40 <= 50 : True.

i = i+1, i == 6.

L[i] ↔ L[i], L[6] ↔ L[8]



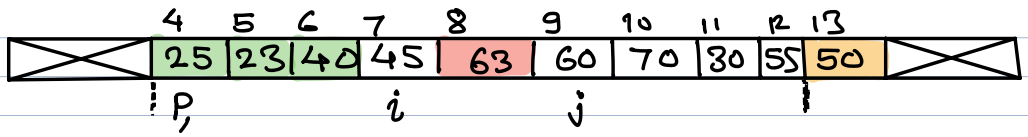
j = j + 1



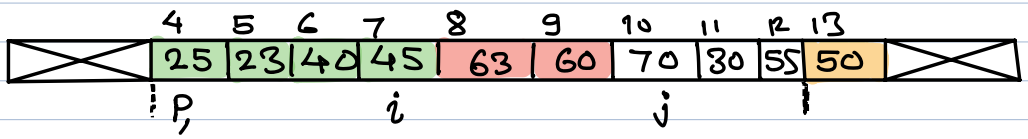
j = 9, i = 6, pivot = 50, L[j] <= pivot, 45 <= 50, True.

i = i + 1, i = = 7

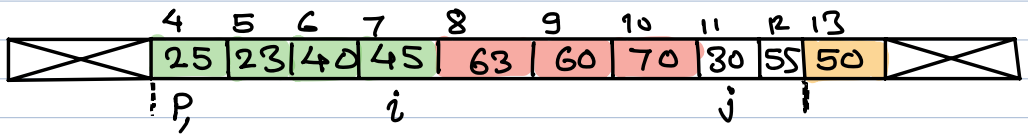
L[i] ↔ L[j] L[7] ↔ L[9]



j = j + 1



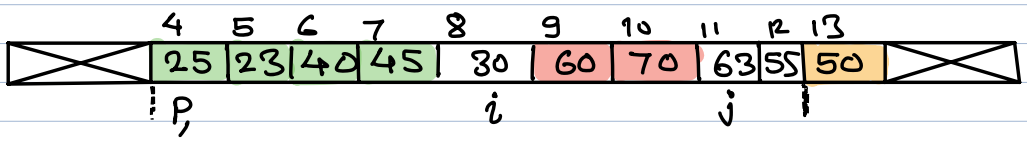
j = 10, i = 7, pivot = 50, L[j] <= pivot, 70 <= 50 : false
j = j + 1, Go to next



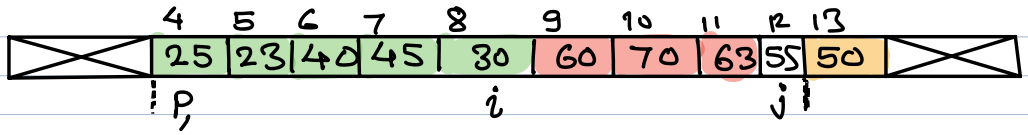
j = 11, i = 7, pivot = 50, L[j] <= pivot, 30 <= 50 : True.

i = i + 1 , i = = 8

L[i] ↔ L[j], L[8] ↔ L[11]

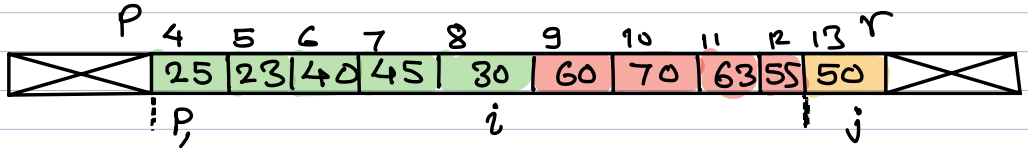


$j = j + 1$

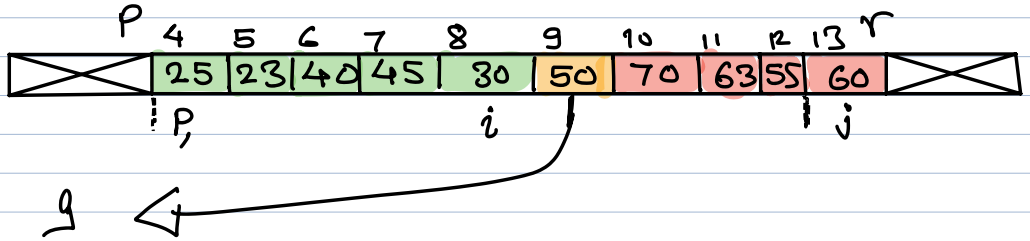


$j = 12$, $i = 8$, pivot = 50, $L[j] <= pivot$, $55 <= 50$, False

$j = j + 1$, Goto next ⟶ STOP.



$L[i+1] \longleftrightarrow L[r]$



Loop - Invariant:

$\forall k \in [p, i]$, $L[k] \leq pivot$

$\forall k \in (i, j)$, $L[k] > pivot$.

---

formal verification: David Gries, Dijisktra, Tony Hoare

Prop. logic, Predicate logic, Set theory, Induction.

---

loop-variant : kernel