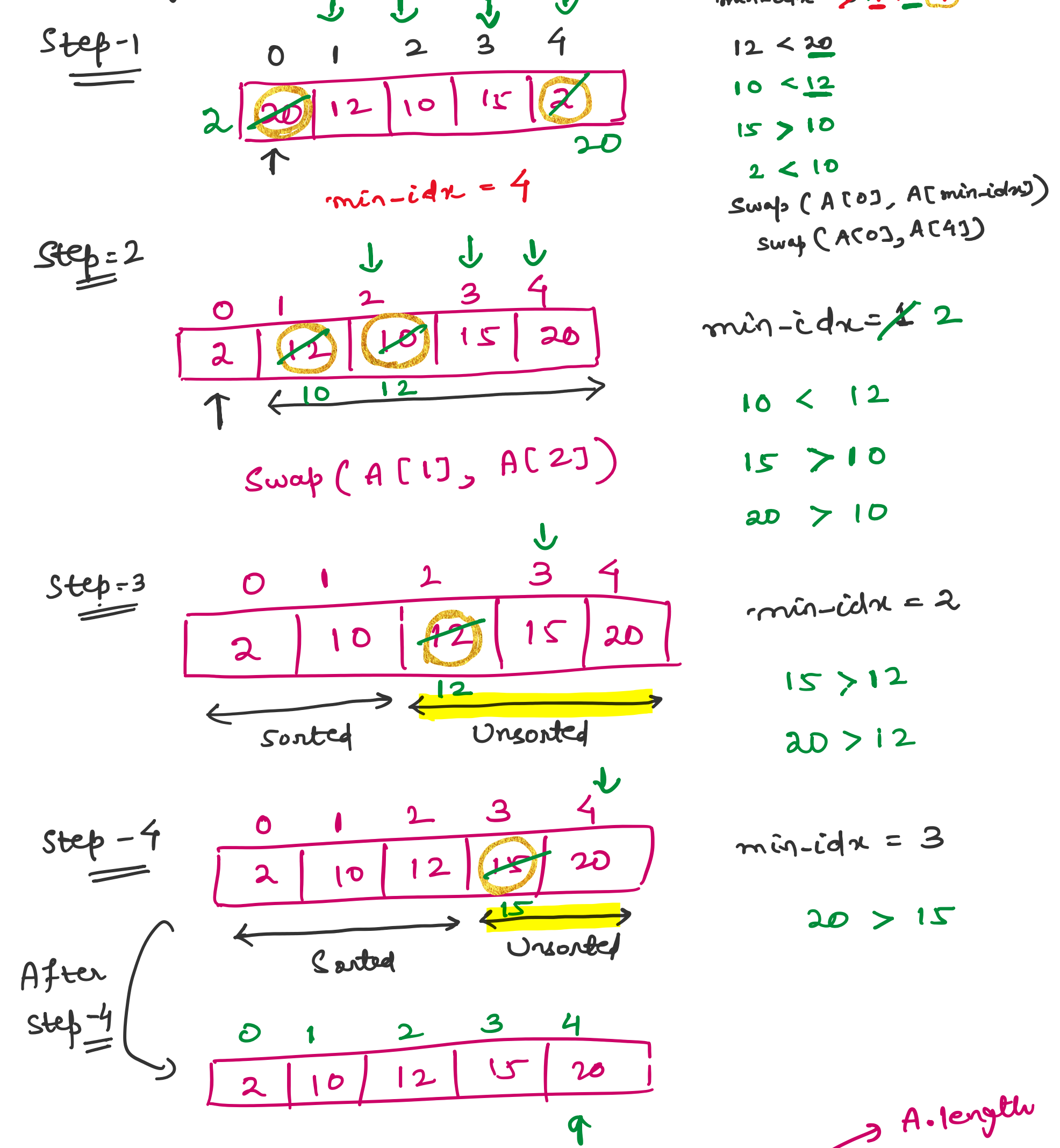


1 → select the smallest element from unsorted Array & place it at beginning of unsorted list.

2 → select the largest element from unsorted Array & place it at end of unsorted list.



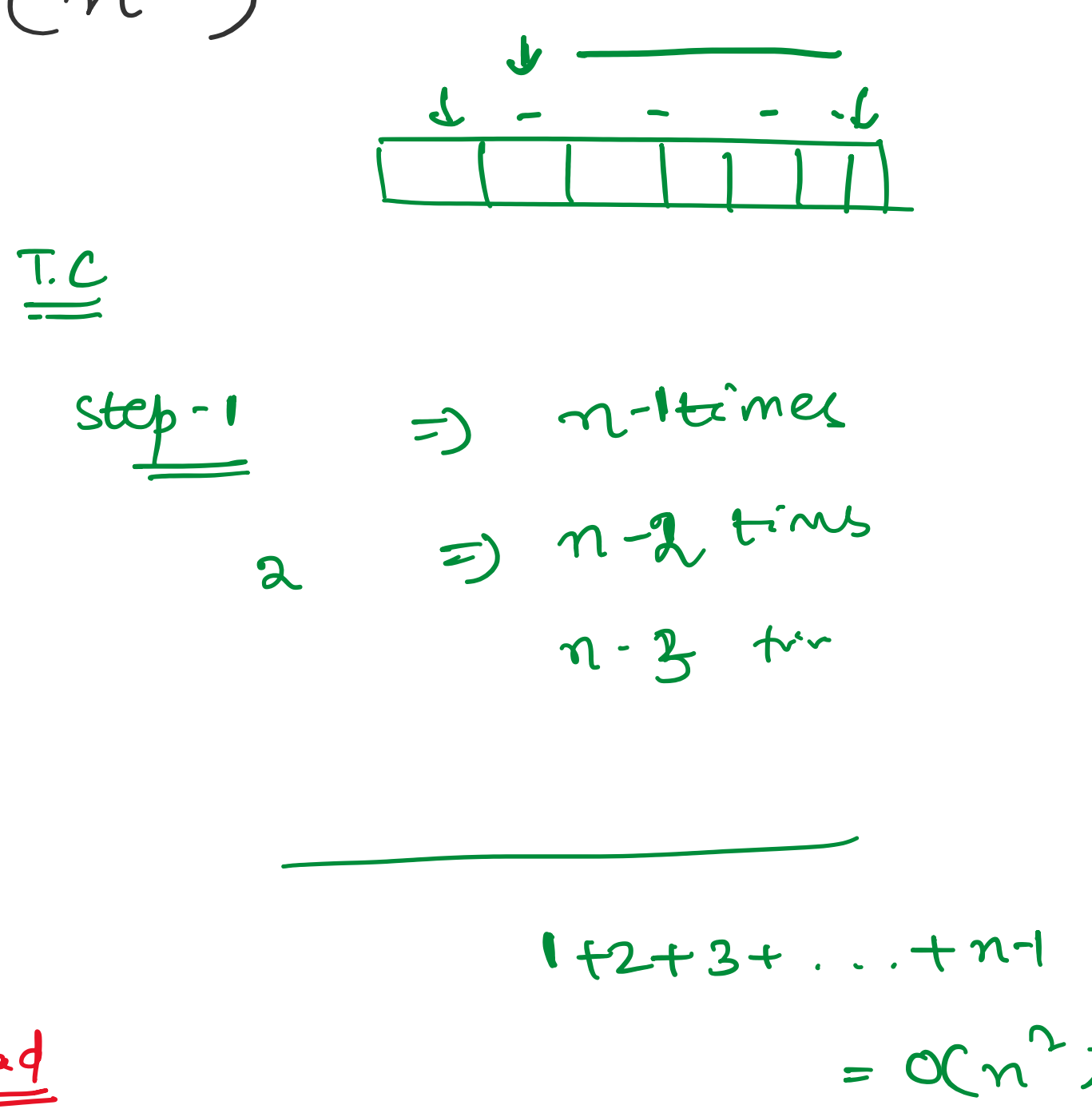
```
function selectionSort(A, n) {  
  let n = A.length; let temp, min-idx;  
  for(i = 0; i < n-1; i++) {  
    min-idx = i;  
    for(j = i+1; j < n; j++) {  
      if(A[j] < A[min-idx]) {  
        min-idx = j;  
      }  
    }  
    swap {  
      temp = A[i];  
      A[i] = A[min-idx];  
      A[min-idx] = temp;  
    }  
  }  
}
```

T.C = $O(n \times n) = O(n^2)$

S.C = $O(1)$

Best case
Worst case
Average case

⇒ $O(n^2)$



X Bubble Sort

Each iterations, we have $(n-i)$ swaps.

swaps = $(n-1) + (n-2) + \dots + 1$

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

$= \frac{n^2 - n}{2}$

B.C ⇒ $O(n)$
W.C ⇒ $O(n^2)$
A.C ⇒ $O(n^2)$

Selection Sort

Each iterations, we have just one swap.

of swaps = $(n-1)$

In case write heavy;
selection sort is good.

B.C ⇒ $O(n^2)$
W.C ⇒ $O(n^2)$
A.C ⇒ $O(n^2)$

