W10 PCRS

Linear Search

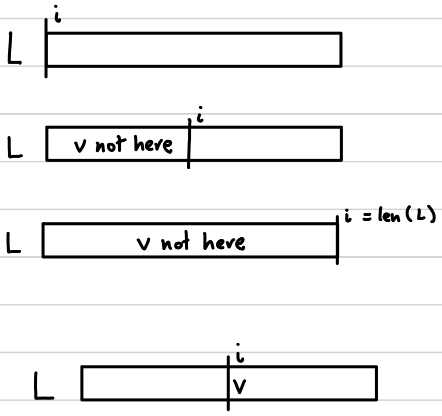
```
def  linear_search(L : list, v : object) -> int :
    i = 0
    while  i != len(L)  and   v != L[i]:
        i =  i + 1
    if  i == len(L):
        return  -1
    else:
        return  i
```

Initial state:

General case:

After the loop terminates, if v isn't in the list :

After the loop terminates, if v is in the list:



Invariant : Something that doesn't change.

Loop invariant : Something that is true before and after each iteration of the loop.

Binary Search

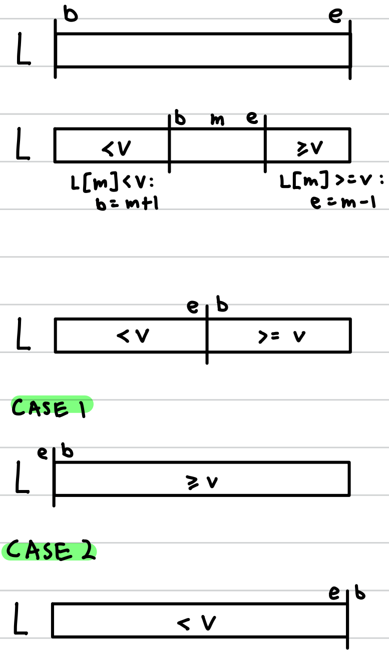
```
def  binary_search(L: list, v:  object) -> int:
    b = 0
    e = len(L) - 1
    while  b <= e :
        m = (b + e) // 2
        if  L[m] < v :
            b = m + 1
        else :
            e = m - 1
    if  b == len(L)  or  L[b] != v :
        return  -1
    else :
        return  b
```

Initial State:

General State:

After loop terminates, v is in L :

After loop terminates, v is not in L :



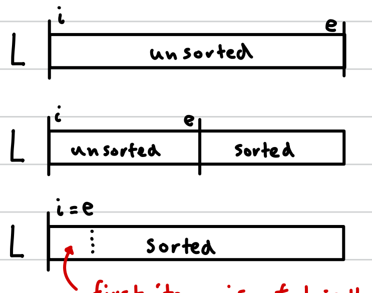
↳ Requires list to be sorted first.

Bubble Sort

Initial State:

General State:

End State:



first item is technically unsorted, but it is smaller than the rest.

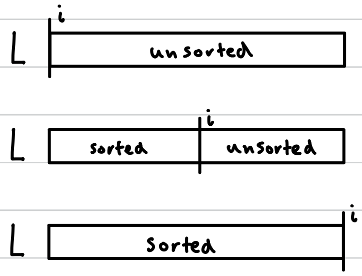
```
def  bubble_sort (L: List) -> NoneType:
     end = len(L) -1
     while  end != 0 :
            for  i  in  range(end):
                 if  L[i] > L[i+1]:
                     L[i], L[i+1]= L[i+1], L[i]
            end = end -1
```

Selection Sort

Initial State:

General State:

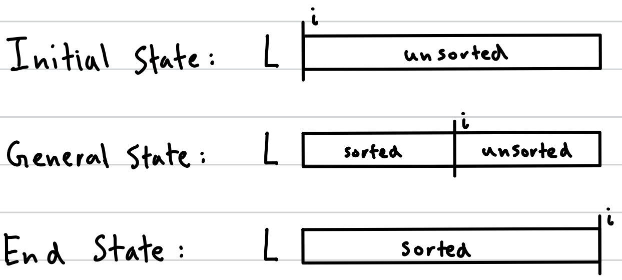
End State:



```
def  get_index_of_smallest (L: list, i: int) -> int:
     index_of_smallest = i
     for j  in  range ( i + 1 , len(L)):
         if  L[j] < L[ index_of_smallest]:
             index_of_smallest = j
     return  index_of_smallest


def  selection_sort (L: list) -> None:
     for  i  in  range (len (L)):
          index_of_smallest = get_index_of_smallest (L,i)
          L[index_of_smallest] , L[i] = L[i], L[index_of_smallest]
```

Insertion Sort



```
def  insert ( L: list, i : int) -> None:
     value = L[i]
     j = i
     while  j != 0  and  L[j-1] > value:
          L[j] = L[j -1]
          j = j -1
     L[j] = value

def  insertion_sort ( L: list) -> None:
     for  i  in  range(len(L)):
          insert (L, i)
```