Linaer search and binary search

size = 10 #...........Size of the list

#...........List with Random/Unordered Values

random\_List = [ 78, 35, 56, 28, 98, 67, 25, 54, 12, 9]

#...........List with Sorted/Ordered Values

sorted\_List = [ 9, 12, 25, 28, 35, 54, 56, 67, 78, 98]

#.................Function to find Key using Linear Search...................

print("\n ................Function 01: To find Key using Linear Search................")

def linear\_Search():

print("\n ...............Linear Search Algorithm................")

print("\n List: ", random\_List)

key = int(input("\n Enter a Key Value: "))

index = 0

comp = 0

for x in random\_List:

if x == key:

comp += 1

print("\t Student have Attended Training .....!!!")

print("\t At Index: ", index)

print("\t Comparisons: ", comp)

break

else:

index += 1

comp += 1

if(index == size):

print("\t Student Not Attended Training .....!!!")

print("\t Comparisons: ", comp)

#.................Function to find Key using Sentinel Search...................

print("\n ................Function 02: To find Key using Sentinel Search................")

def sentinel\_Search():

print("\n ...............Sentinel Search Algorithm................")

print("\n List: ", random\_List)

last = random\_List[size-1]

key = int(input("\n Enter a Key Value: "))

random\_List[size-1] = key

index = 0

comp = 0

for x in random\_List:

if x == key:

comp += 1

break

else:

index += 1

comp += 1

if(comp < size or key == last):

print("\t Key Found .....!!!")

print("\t At Index: ", index)

print("\t Comparisons: ", comp)

else:

print("\t Key Not Found .....!!!")

print("\t Comparisons: ", comp)

#.................Function to find Key using Binary Search...................

print("\n ................Function 03: To find Key using Binary Search................")

def binary\_Search():

print("\n ...............Binary Search Algorithm................")

print("\n List: ", sorted\_List)

key = int(input("\n Enter the Roll No:"))

low = 0

high = len(sorted\_List) - 1

mid = 0

comp = 0

while low <= high:

mid = int((low + high) / 2)

if key == sorted\_List[mid]:

comp += 1

print("\t Key Found")

print("\t At Index = ",mid)

print("\t Comparisons = ", comp)

break

elif key < sorted\_List[mid]:

high = mid - 1

else:

low = mid + 1

comp += 1

if low > high:

print("\t Key Not Found")

print("\t Comparisons = ", comp)

#.................Function to find Key using Fibonacci Search...................

print("\n ................Function 04: To find Key using Fibonacci Search................")

#.......Fibonacci Series

fibo = [ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

def fibonacci\_Search():

print("\n ...............Fibonacci Search Algorithm................")

print("\n Sorted List: ", sorted\_List)

print("\n Fibonacci Series: ", fibo)

key = int(input("\n\t Enter Roll No to search: "))

k = 0 #....... To find fibo(k) >= Size of Sorted List

while fibo[k] < len(sorted\_List):

k += 1

#....... Offset = -1

offset = -1

comp = 0

while k > 0: #.......Find index i = min((offset + fibo[k-2]) , size-1)

i = min((offset + fibo[k-2]) , len(sorted\_List)-1)

if key == sorted\_List[ i ]:

comp += 1

print("\t Student Have Attended Training Program.")

break

elif key > sorted\_List[ i ]:

k = k-1

offset = i

else:

k = k - 2

comp += 1

if k <= 0:

print("\t Student Have Not Attended Training Program.")

print("\t Comparisons Needed = ", comp)

#...................Menu Driven Programming

choice = 0

while(choice <= 4):

print("\n--------\*\* MENU \*\*--------")

print("\t 1. Linear Search")

print("\t 2. Sentinel Search")

print("\t 3. Binary Search")

print("\t 4. Fibonacci Search")

print("\t 5. STOP/EXIT")

choice = int(input("\n\t Enter your Choice(1/2/3/4): "))

if choice == 1:

linear\_Search()

elif choice == 2:

sentinel\_Search()

elif choice == 3:

binary\_Search()

elif choice == 4:

fibonacci\_Search()

elif choice ==5:

print("Thank you")

quit()

else:

print("Enter valid choice")