
Started on Friday, 24 January 2025, 3:32 PM

State Finished

Completed on Friday, 24 January 2025, 3:59 PM

Time taken 27 mins 29 secs

Grade 100.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement merge sort using iterative approach on the given list of values.

For example:

Test	Input	Result
Merge_Sort(S)	6 4 2 3 1 6 5	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]
Merge_Sort(S)	5 2 6 4 3 1	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]

Answer: (penalty regime: 0 %)

```

1 def merge(S, temp, From, mid, to):
2     a = From
3     b = From
4     c = mid + 1
5     while b <= mid and c <= to:
6         if S[b] < S[c]:
7             temp[a] = S[b]
8             b = b + 1
9         else:
10            temp[a] = S[c]
11            c = c + 1
12            a = a + 1
13     while b < len(S) and b <= mid:
14         temp[a] = S[b]
15         a = a + 1
16         b = b + 1
17     for b in range(From, to + 1):
18         S[b] = temp[b]
19 def Merge_Sort(S):
20     low = 0
21     high = len(S) - 1
22     temp = S.copy()

```

	Test	Input	Expected	Got	
✓	Merge_Sort(S)	6 4 2 3 1 6 5	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]	✓

	Test	Input	Expected	Got	
✓	Merge_Sort(S)	5 2 6 4 3 1	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]	✓
✓	Merge_Sort(S)	4 3 5 6 1	The Original array is: [3, 5, 6, 1] Array after sorting is: [1, 3, 5, 6]	The Original array is: [3, 5, 6, 1] Array after sorting is: [1, 3, 5, 6]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 2

Correct

Mark 20.00 out of 20.00

Write a python program to implement quick sort on the given float values and print the sorted list and pivot value of each iteration.

For example:

Input	Result
5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]
4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]

Answer: (penalty regime: 0 %)

```

1 def quick_sort(alist, start, end):
2     if end - start > 1:
3         p = partition(alist, start, end)
4         quick_sort(alist, start, p)
5         quick_sort(alist, p + 1, end)
6 def partition(alist, start, end):
7     pivot = alist[start]
8     i = start + 1
9     j = end - 1
10    print("pivot: ",pivot)
11    while True:
12        while (i <= j and alist[i] <= pivot):
13            i = i + 1
14        while (i <= j and alist[j] >= pivot):
15            j = j - 1
16        if i <= j:
17            alist[i], alist[j] = alist[j], alist[i]
18        else:
19            alist[start], alist[j] = alist[j], alist[start]
20        return j
21    alist = []
22    n=int(input())

```

	Input	Expected	Got	
✓	5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	✓

	Input	Expected	Got	
✓	4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	✓
✓	6 3.1 4.2 5.1 2.3 7.4 5.9	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a Python Program Using a recursive function to calculate the sum of a sequence**For example:**

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

```

1 def sum(n):
2     if(n==1):
3         return n
4     else:
5         return n+sum(n-1)
6 n=int(input())
7 result=sum(n)
8 print(result)

```

	Input	Expected	Got	
✓	20	210	210	✓
✓	36	666	666	✓
✓	45	1035	1035	✓
✓	58	1711	1711	✓
✓	65	2145	2145	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 4

Correct

Mark 20.00 out of 20.00

Write a python program to implement linear search on the given tuple of float values.

note: As the tuple is immutable convert the list to tuple to perform search

For example:

Input	Result
5 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found
6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	Tuple: 6.2 found

Answer: (penalty regime: 0 %)

```

1 def LS(list,n,key):
2     for i in range(0,n):
3         if(list[i]==key):
4             return i
5     return -1
6 list=[]
7 n=int(input())
8 for i in range(0,n):
9     temp=input()
10    list.append(temp)
11 key=input()
12 res=LS(list,n,key)
13 if(res==-1):
14     print("Tuple: %s not found"%key)
15 else:
16     print("Tuple: %s found"%key)

```

	Input	Expected	Got	
✓	5 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found	Tuple: 6.4 found	✓

	Input	Expected	Got	
✓	6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	Tuple: 6.2 found	Tuple: 6.2 found	✓
✓	4 2.1 3.2 6.5 4.5 3.5	Tuple: 3.5 not found	Tuple: 3.5 not found	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program to implement binary search on the given list of string values using iterative method

For example:

Test	Input	Result
binarySearchAppr(arr, 0, len(arr)-1, x)	5 one two three four five two	Element is present at index 4
binarySearchAppr(arr, 0, len(arr)-1, x)	6 one three five seven nine eleven thirteen	Element is not present in array

Answer: (penalty regime: 0 %)

```

1 def binarySearchAppr(arr, start, end, x):
2     if end >= start:
3         mid = (start + end) // 2
4         if arr[mid] == x:
5             return mid
6         elif arr[mid] > x:
7             return binarySearchAppr(arr, start, mid-1, x)
8         else:
9             return binarySearchAppr(arr, mid+1, end, x)
10    else:
11        return -1
12 arr = []
13 n = int(input(""))
14 for i in range(n):
15     arr.append(input())
16 arr = sorted(arr)
17 x = input()
18 result = binarySearchAppr(arr, 0, len(arr)-1, x)
19 if result != -1:
20     print(f"Element is present at index {result}")
21 else:
22     print("Element is not present in array")

```

	Test	Input	Expected	Got	
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	5 one two three four five two	Element is present at index 4	Element is present at index 4	✓

	Test	Input	Expected	Got	
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	6 one three five seven nine eleven thirteen	Element is not present in array	Element is not present in array	✓
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	4 two four six eight six	Element is present at index 2	Element is present at index 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.