

Homework 6

ECE 590: Towards More Reliable Software

Jeff Fan || zf70@duke.edu
February 20, 2024

Question 1:

```
1 def quicksort(arr):
2     if len(arr) <= 1:
3         return arr
4     pivot = arr[len(arr) // 2]
5     left = [x for x in arr if x < pivot]
6     middle = [x for x in arr if x == pivot]
7     right = [x for x in arr if x > pivot]
8     return quicksort(left) + middle + quicksort(right)
9
10 def insertion_sort(arr):
11     for i in range(1, len(arr)):
12         key = arr[i]
13         j = i-1
14         while j >= 0 and key < arr[j] :
15             arr[j + 1] = arr[j]
16             j -= 1
17         arr[j + 1] = key
18     return arr
19
20 def is_sorted(arr):
21     return all(arr[i] <= arr[i+1] for i in range(len(arr)-1))
22
23 def sort_with_recovery(arr):
24     print("Attempting primary variant (QuickSort)")
25     sorted_arr = quicksort(arr)
26     if is_sorted(sorted_arr):
27         print("Primary variant succeeded.")
28         return sorted_arr
29     else:
30         print("Primary variant failed. Attempting alternate variant (InsertionSort)")
31         sorted_arr = insertion_sort(arr)
32         if is_sorted(sorted_arr):
33             print("Alternate variant succeeded.")
```

```

34     return sorted_arr
35 else:
36     print("All variants failed.")
37     raise Exception("Recovery failed.")
38
39 arr = [3, 2, 5, 1, 6, 4]
40 sorted_arr = sort_with_recovery(arr)
41 print("Sorted array:", sorted_arr)

```

This code implements a recovery block scheme using two sorting algorithms, QuickSort and InsertionSort, to sort an array. QuickSort is used as the primary variant, while InsertionSort serves as the alternate variant.

If QuickSort Succeeds: The program begins by attempting to sort an array using QuickSort. If this primary variant successfully sorts the array, as verified by the `is_sorted` function, the program outputs the following:

```

(ECEbase) yume@Zhes-MacBook-Pro code % /Users/yume/Env/ECEbase/bin/python3 /Users/yume/Desktop/DUKE/Curricular/24Spr/590/hwk6/code/recoveryBlock.py
Attempting primary variant (QuickSort)
Primary variant succeeded.
Sorted array: [1, 2, 3, 4, 5, 6]

```

This indicates that QuickSort was able to sort the array without any issues, and there was no need to attempt sorting with the alternate variant.

If QuickSort Fails and InsertionSort Succeeds: We can change line 5 to be:

```

1 left = [x for x in arr if x > pivot]

```

If the primary variant (QuickSort) fails to sort the array correctly, the program proceeds with the alternate variant (InsertionSort). If InsertionSort then successfully sorts the array, the output is:

```

(ECEbase) yume@Zhes-MacBook-Pro code % /Users/yume/Env/ECEbase/bin/python3 /Users/yume/Desktop/DUKE/Curricular/24Spr/590/hwk6/code/recoveryBlock.py
Attempting primary variant (QuickSort)
Primary variant failed. Attempting alternate variant (InsertionSort)
Alternate variant succeeded.
Sorted array: [1, 2, 3, 4, 5, 6]

```

This output demonstrates the recovery block scheme's ability to revert to an alternate sorting method upon the failure of the primary method, ensuring the task's completion.

If Both QuickSort and InsertionSort Fail: We can change line 14 to be:

```

1 while j == 0 and key < arr[j] :

```

In the unlikely event that both QuickSort and InsertionSort fail to sort the array correctly, the program will raise an exception to indicate the failure of the recovery process, with the output being:

```
(ECEbase) yume@Zhes-MacBook-Pro code % /Users/yume/Env/ECEbase/bin/python3 /Users/yume/Desktop/DUKE/Curricular/24Spr/590/hwk6/code/recoveryBlock.py
Attempting primary variant (QuickSort)
Primary variant failed. Attempting alternate variant (InsertionSort)
All variants failed.
Traceback (most recent call last):
  File "/Users/yume/Desktop/DUKE/Curricular/24Spr/590/hwk6/code/recoveryBlock.py", line 40, in <module>
    sorted_arr = sort_with_recovery(arr)
  File "/Users/yume/Desktop/DUKE/Curricular/24Spr/590/hwk6/code/recoveryBlock.py", line 37, in sort_with_recovery
    raise Exception("Recovery failed.")
Exception: Recovery failed.
```

This indicates a complete failure of the sorting process, where neither the primary nor the alternate sorting methods were able to produce a correctly sorted array, leading to the raising of an exception.