CSE221

Lab Assignment-3

Name : Fathin Ishrak

ID: 2030102x

Section: 13

Task 1

This code reads the input data from "input1.txt", where the first line represents the number of elements "n", and the second line containg the space-seperated list of numbers. The "merge" function merges two sorted lists "a" and "b" into single sorted list, and it uses two pointers "i" and "j" to iterate through the lists and compare elements. The "mergeSort" function recursively divides the input list "arr" into smaller sublists until they are reduced to a size of 1 or less. It then merges these sublists using the "merge" function to produce a sorted result. After sorting the numbers using Merge Sort algorithm, it writes the sorted result to an output file named "output1.txt", with each number seperated by a space.

Task2

This code reads input data from "input2.txt", where the first line represents the number of elements "n" and the second line containg the space seperated list of numbers. The function "findMax" recursively finds the maximum value in an array "arr" within a given range

Scanned with CamScanner

defined by the "left" and "right" indices. Finally it returns the maximum value among the left and right halves in an output file named "output2.txt" using the "main" function.

The time complexity of this code is O(n) because it recursively ~~divide~~ divides the list into two halves and computes the maximum value for each half, resulting in a linear number of comparisons.

Task 3

This code reads input data from "input3.txt", where the first line represents the total number of aliens (1 <= n <= 10^6) and the second line contains the space-seperated list of heights. The "merge_and_~~cont~~ count_inversions" function takes an array "arr" and the indices "left", "mid", and "right" to merge to subarrays and count the inversions. It divides the array into left and right subarrays, compares elements, and counts inversions while merging. The "count_inversions_and_sort" function uses a modified merge sort algorithm to recursively divide the array into smaller subarrays and counts inversions. It combaines the counts from both halves and inversions during merging. Finally writes the result to an output file named "output3.txt."

Scanned with CamScanner

Task 4

This code reads the data from "input4.txt" file, where the first line containing the length of the list ($1 <= n <= 10^6$). and second line containing n integers seperated by a space. It calculates the maximum possible value of $A[i] + A[j]^2$, where i and j are indices ($1 <= i < j <= n$). It filters out empty strings in the input list, finds the maximum value using a divide-and conquer approach and stores the result in the variable "result". It then writes the output to "output4.txt" while iterating through the list and updating the maximum value accordingly.

Scanned with CamScanner

Task 5

The "partition" function selects a pivot element from the list and rearranges the elements so that all values less than or equal to the pivot are on the left, and all values greater than the pivot are on the right. It returns the index of the pivot. The "quickSort" function recursively divides the list into smaller sublists and sorts them using the "partition" function. It continues this process until the entire list is sorted.

Task6

The "quick_select" function recursively selects a pivot element from the middle of the list and divides the list into 3 parts: "left" (elements less than the pivot), "middle" (elements equal to the pivot) and "right" (elements greater than the pivot). It checks the value of "k" to determine in which part the k-th smallest element exists. If "k" is less than the number of elements in the "left" part, it recursively calls the function on the "left" part. If "k" falls within the "middle" part, it returns the pivot value (the k-th smallest). Otherwise it recursively calls the function on the "right" part while adjusting "k" accordingly. This code reads input from "input6.txt", processes multiple queries to find the k-th smallest values, and writes the result to "output6.txt" for each query.

Scanned with CamScanner