Seoul National Uiniversity

M1522.000900 Data Structure

Homework 6: Internal Sorting (Chapter 7)

Computer Science & Engineering

2017-18538 Hwang Sun Young

Question1

(1) [0, 1, 2, 4, 5, 7, 8]

(2) The time complexity of the algorithm is Θ(n+k)

(3) Let the size of input array and aux array be n and the size of counts array be k. The time space of the algorithm is Θ(n+k)

(4) because in this algorithm the original ordering for duplicate keys is preserved, this algorithm is stable.

Question2

n = 2^m-1 merge sort

1. Call it once for input array. We can get 2 sub-arrays.
2. Call it twice for 2 arrays. We can get 4 sub-arrays.
3. Call it four times for 4 arrays. We can get 8 sub-arrays. And so on….

1+2+4+…+2^{m-1} = 2^m-1

The number of last call is 2^{m-1}. It calls insertion sort.

So insertion sort is called 2^{m-1} times for 2^m-1 calls in merge sort.

Question3

If the ratio of the two sub-lists is always 1:99, one of them is sorted already and another have to be sorted. Let the size of input list be n. we can divide list(ratio of the two sub-lists is always 1:99).

Then we have to sort sub-list(it charge 99 of ratio). This process goes on recursively. So the time complexity is n+(n-1)+...+2+1 = n(n+1)/2 and O(n^2).

Question4

TAB BAR EAR TAR SEA TEA DIG BIG MOB DOG COW ROW NOW BOX FOX RUG

Question5

(1) The pair of array element.

(2,1), (3,1), (8,6), (8,1), (6,1)

(2) (n, n-1, ..., 3, 2, 1) This array has the largest number of inversions.

(n-1) + (n-2) + ... + 2 + 1 = (n-1)n/2

It has (n-1)n/2 inversions.

(3) 1 swap decrease the number of inversions by 1. For each i, the number of comparisons is equal to, or one greater than the number of swaps. Let the number of comparisons of insertion sort be n and the number of inversions in the input array be d.

d <= n <= d + (n-1)