1. Master’s theorem can be used to resolved this: T(n/b) +f(n) where, T(n/b) should be in form of n/b, but here it is (n-1)/b

So, Algebraic substitution is used where T (1) = 1 is given.

T (2) = T (2-1) + ½ = T (1) + ½ = 3/2

T (3) = T (3-1) + 1/3 = T (2) + 1/3 = 3/2 + 1/3 = 11/6

T (4) = T (4-1) + 1/4 = T (3) + 1/4 = 11/6 + 1/4 = 25/12

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

T(n) = T(n-1) + (2n-1) / (n2- n)

There for, T(n) = Log (n)

1. For the notation that f(n) is Ω(n)

f(n) = Ω(n) if and only if lim n→∞ [f(n)/g(n)]>0

The definition of the Big Omega is the function f(n) is Ω(g(n)) iff there exists a positive real constant c and a positive integer n0 such that f(n) ≥ cg(n) for all n > n0

F(n) = n2 +10 n, g(n) = n2

N2 +10 n ≥ C\*n2

Assume n0 = 1;

c ≤ 1 + 10,

c ≤ 11.

According to the definition c > 0 and n0 ≥ 1,

We have n2 +10 n = Ω(n2)

This equation can be written as f(n) = n2 +10n ≥ n2, where n ≥ 1

Therefore, f(n2) = Ω(n2).

1. Yes, sorting algorithm can be modified to make them stable. There are different methods to make the sorting algorithm stable but the best way is when comparison based sorting algorithm is not stable by nature it can be modified by changing the key comparison operation where both the keys considers position as a factor for objects with equal keys to make it stable.
   1. Let’s assume radix sort here an element is found that has the same key it is inserted to the list of similar keys. So, when the sorting is executed the list of keys is ready and in principal list keys are in sorted order where every node have another list while they retain the objects in their occurrence. At the end main list is read in the default arrangement which would get all the elements in a sorted order.
   2. Selection sort can be implemented where the exchange in the array happens at end of each sorting round which would change the values in relation to the location and order of the data.

Example: unsorted numbers- 4 2 3 4 1

Here the order of number 4 is change the 1st 4 in the array is shifted to the location after the second 4. Hence, 1 2 3 4 4

Let’s try a simulation of another numbers

Smallest value is searched - 2 3 1 1

Smallest value is exchanged with the value at the key 0 – 1 2 3 1

The smallest value is searched from keys 1 to n as 0 is already changed to smallest – 1 3 2 1

The smallest value is position at 1 and other elements are moves and this goes on – 1 1 2 3

1. add method: with the help of accounting method the potential from data structure is released to pay for future operations. The same potential is like the considering total credit stored in entire data structure. Now, once the capacity is confirmed that it is full an element is added in the array with the charge of 2 cyber dollars, and if the array is at its full capacity then the array where profits have been overcharged on some operations is doubled. Those profits execute primitive operations and the profit assist other operation to execute many other primitive operations.

remove method: in this method if the size is less than n/4 then as the given condition the array is supposed to be shrink.

So, to shrink the deference of N/2 elements is required to remove them from the array, which would result in the time complexity of O(n/2) of course here the constants are ignored and the time complexity is resulted to O (n)

1. Merge sort is implemented which has the time complexity of n log n. Then from throughout the array we can traverse from 0 to size n -1 which if results X from first element of the array