Session-2

We will use a normal array, and a var 'len' to store its length.

Insert(x): simply append x to the array. Note that length of array increases by 1 ie (len++)

Delete-Larger-Half(S)

Find the median of the array.

Now, divide the array such that upper half is stored in the last $|S|/2$ ~~elem~~ positions. Delete these elements, hence size of array is halved. (len/=2)

The time complexity of Insert Opⁿ is $O(1)$ and Delete-Larger-Half is $O(n)$.

The potential function $\phi(S) = c\,2|S|$ can be used to show that amortized cost of any Insert or Delete-Larger-Half operation is $O(1)$. Clearly ϕ satisfy the necessary conditions. Here c is constant time, as taken in class

A.C → Actual cost

In case of Insert, $\Delta\phi = 2c$, thus amortised cost for Insert is $A.C + \Delta\phi = c + 2c = 3c$

In case of Delete-~~Regular~~ Larger-Half, $2\left(\frac{n}{2}\right)c - 2(n)c = -nc$

hence amortised cost for this is $A.C + \Delta\phi = 0$.

Actual cost

Hence, amortised cost proved $O(1)$ for each.

In order to print all elements, simply output all the elements in the array. → $O(|S|)$