2. Answer:

(i) VEB-EMPTY-TREE-INSERT (V, x)

- This procedure does not change because the original algorithm from CLRS3 already function with the case of maximum member of the universe set (V.max) and is correct as per new requirement.

```
V.min = x
V.max = y
```

(ii) VEB-TREE-INSERT (V, x)

- Here we can handle V.max in similar terms as V.min is handled. Following is the modified algorithm;

```
if V.min == NIL then
    VEB-EMPTY-TREE-INSERT (V, x)
else
    if x < V.min then
        exchange x with V.min
    end if
    if x > V.max then
        exchange x with V.max
    end if
    if V.u > 2
        if VEB-TREE-MINIMUM (V.cluster [high (x)]) == NIL
            VEB-TREE-INSERT (V.summary, high (x))
            VEB-EMPTY-TREE-INSERT (V.cluster [high (x)],
                                                 low (x))
        else VEB-TREE-INSERT (V.cluster [high(x)], low (x))
```

2 (iii) VEB-TREE-DELETE (V, x) (line 9-12)

- Need to handle V.max in similar way as V.min is carried out in original algorithm. Following is the updated algorithm:

→ VEB-TREE-DELETE (V, x)

```
1.  if V.min == V.max
2.      V.min = NIL
3.      V.max = NIL
4.  else if V.u == 2
5.      if x == 0
6.          V.min = 1
7.      else if V.min = 0
8.          V.max = V.min
9.  else if x == V.min
10.         first-cluster = VEB-TREE-MINIMUM (V.summary)
11.         x = index ( first-cluster, VEB-TREE-MINIMUM
                    (V. cluster [first-cluster]) )
12.         V.min = x
13. if x == V.max then
14.         last-cluster = VEB-TREE-MAXIMUM (V.summary)
15.         x = index ( last-cluster, VEB-TREE-MAXIMUM
                    (V. cluster [last-cluster]) )
16.         V.max = x
17. VEB-TREE-DELETE (V. cluster [high(x)], low(x))
18. if VEB-TREE-MINIMUM (V.cluster [high (x)]) == NIL
19.     VEB-TREE-DELETE (V.summary, high (x))
```

→ VEB-TREE-MINIMUM (V) → return V.min

→ VEB-TREE-MAXIMUM(V) → return, V.max

3. (a) Consider universe set has u elements

$\therefore$ By divide and conquer, we can divide universe into $\sqrt{u}$ clusters

- Their will exist 1 summary for maintaing minimum, maximum, etc values of cluster belonging to u. set
- Therefore, space required for single cluster

$$P(\sqrt{u}) \quad \text{—— (1)}$$

- $\Theta(\sqrt{u})$ space will be required for storing array of pointers, size, minimum & maximum values —— (2)
- $$\therefore P(u) = (\sqrt{u}+1) P(\sqrt{u}) + \Theta(\sqrt{u})$$

— from (1) & (2)

(b) Consider total space required by universe set u as $P(u) = c(u-2)$

$\therefore$ Each cluster will need $P(\sqrt{u}) = c(\sqrt{u}-2)$ space.

- Considering $P(u)$ equation from (a), we get

$$P(u) = (\sqrt{u}+1) P(\sqrt{u}) + \Theta(\sqrt{u})$$

$$= (\sqrt{u}+1) \cdot c(\sqrt{u}-2) + \Theta(\sqrt{u})$$

3. (b) $P(u) = c(u - 2\sqrt{u} + \sqrt{u} - 2) + \Theta(\sqrt{u})$

$= c(u - \sqrt{u} - 2) + \Theta(\sqrt{u})$

$= c(u-2) - c\sqrt{u} + \Theta(\sqrt{u})$

if value of c would be larger, then $-c\sqrt{u}$ & $\Theta(\sqrt{u})$ will be cancelled out.

$\therefore \quad P(u) \le c(u-2)$

i.e. $O(c(u-2)) = O(u)$

$\therefore \quad P(u) = O(u) \longrightarrow$ proved

(c) with the same assumption as in (b), we have

$P(u) = (\sqrt{u}+1) \cdot c(\sqrt{u}-2) + O(1)$

$= c(u-2) - c\sqrt{u} + O(1)$

$\le c(u-2)$

- if value of 'u' will be larger then ~~constant~~ c can also be evaluated to some value and $c\sqrt{u}$ would be larger than a constan $O(1)$

$\therefore P(u) \le c(u-2)$

- Therefore, moving pointers outside vEB ~~will~~ structure will not improve vEB structure.