Heap A before insert:

(a) Addition of new leaf:

(b) A place for '3' is found:

(c) Procedure returns 15 as the max.

and (d) gives the heapified result.
7.5.5 If \( \text{heap-size}[A] < 1 \)

\[ \text{error flag} = \text{True}. \]

\[
\begin{align*}
A[i] &\leftarrow A[\text{Heapsize}[A]] & \text{O}(1) \\
\text{Heapsize}[A] &\leftarrow \text{Heapsize}[A] - 1 & \text{O}(1) \\
\text{Heapify}[A, i] & \quad \text{O}(\log n)
\end{align*}
\]

\[ \text{ORDER} = \text{O}(1) + \text{O}(1) + \text{O}(\log n) = \text{O}(\log n). \]

7.1 a) Counter example which shows that \text{Build-Heap}(A) and \text{Build-Heap'}(A) need not produce the same result all the time.

Take \( \{1, 3, 5, 2\} \)

\[ \begin{array}{c}
\text{Build-heap} \\
\begin{array}{c}
\begin{array}{c}
1 \\
3 \\
2
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
1
\begin{array}{c}
3 \\
5
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
5 \\
3
\begin{array}{c}
1
\end{array}
\end{array}
\end{array}
\]

\[ \begin{array}{c}
\text{Build-heap'} \\
\begin{array}{c}
1
\rightarrow
1 \\
3
\rightarrow
1 \\
5
\rightarrow
1 \\
3
\end{array}
\end{array}
\]

Clearly \( A \neq B \)

\[ \therefore \text{They can produce different results.} \]

b) \text{Build-Heap'}(A)

1. \text{Heap-size}[A] \leftarrow 1 \text{ O}(1)
2. for \( i=2 \) to \text{length}[A] \text{ O}(n-1)
3. do \text{Heap-insert}(A, A[i]) \text{ O}(\log n)

\[ \text{Worst case (when all the elements need to be inserted)} = \text{O}(1) + \text{O}(n-1) \times \text{O}(\log n) \]

\[ \text{\~{\text{\~{}}} O(n\log n)}. \]