Aufgabe 2

b)

Lemma:

> Um ein Element in eine Priority-Queue der Länge n einzufügen, benötigt man im schlimmsten Fall $\lceil \log_2 n \rceil$ Schritte.

Beweis:
Eine Priority-Queue ist ein Binärbaum, und ein Binärbaum hat Höhe $h = \lceil \log_2 n \rceil$. Die Operation `Insert` fügt das neue Element an das Ende der Queue, d.h. in die unterste Ebene des Baums, und ruft dann `Increase-Key` auf. Bei `Increase-Key` wird der Knoten in jedem Schritt eine Ebene nach oben geschoben, welches höchstens h-mal passieren kann. Dadurch werden im Worst-Case mindestens $\lceil \log_2 n \rceil$ benötigt.

Zu zeigen:

> Einen Heap mit der `Insert`-Operation einer Priority-Queue aufzubauen, benötig im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$.

Aus o.g. Lemma ergibt sich für eine Folge von n `Insert`-Operationen auf einen leeren Heap im Worst Case folgende Laufzeit L:

$$
\begin{aligned}
L &\geq \sum_{i=1}^{n} \lceil \log_2 i \rceil \\
&> \log_2 \left(\prod_{i=1}^{n} i \right) \\
&= \log_2(n!) = \Theta(n \log n) \\
&\Rightarrow L \in \Omega(n \log n)
\end{aligned}
$$

□