Einführung in die Algorithmik
Assignment 6.2

Nike Pulow, Henri Heyden
stu239549, stu240825

6.2.1

Wir betrachten die Laufzeit von `elem()`. In einer teilsortierten Datenstruktur wie einem Heap wissen wir bei der Suche nach einem spezifischen Wert im Heap nur, dass der Wert, welchen wir suchen sich nicht im Teilbaum, der einem Knoten folgt, dessen Wert höher ist, als der Wert, den wir suchen. Im schlimmsten Fall, wenn der gesuchte Wert nicht im Baum vorhanden ist, kann man mit linearer Laufzeit rechnen. Diese kann etwas verbessert werden, indem man die oben genannte Eigenschaft bei der Implementierung berücksichtigt, fällt allerdings weiterhin in $O(n)$.
Im besten Fall ist das gesuchte Element das erste Element im Heap, also jenes an der Wurzel; in diesem Fall erhalten wir eine Laufzeitkomplexität von $O(1)$, da es genau einen Zugriff benötigt, um das Element zu finden. Selbiges trifft bei unserer Implementierung auch auf das letzte Element im Heap zu.
Im Average Case liegt das gesuchte Element nicht in der letzten Ebene des Heap, da diese nur im Falle einer vollständig ausgefüllten letzten Ebene $\frac{n}{2} + 1$ der Elemente enthält, aber auch weniger Elemente enthalten kann, während alle Ebenen zuvor auf jeden Fall vollständig gefüllt sind. Für die Laufzeitkomplexität bedeutet das, dass nur auf die Hälfte aller Knoten zugegriffen werden muss, sodass das Element innerhalb von $\frac{1}{2} \cdot n$ Zugriffen gefunden wird. Weiterhin gilt allerdings $\frac{1}{2} \cdot n \in O(n)$. Zieht man jetzt noch hinzu, dass im Durchschnitt die Hälfte aller betrachteten Werte an Knoten größer sind, als der Wert, den man sucht, erhält man einen Faktor $\frac{1}{4} \cdot n$, aber auch hier befinden wir uns weiterhin in linearer Laufzeit.
Wir kommen also zu der Konklusion, dass Suchoperationen im Heap nur in linearer Laufzeit realisierbar sind.

Einführung in die Algorithmik

Assignment 6.2

Nike Pulow, Henri Heyden
stu239549, stu240825

6.2.2

Bei der Annahme einer vollständig gefüllten letzten Ebene müssen für `find_max()` exakt $\frac{n}{2} + 1$ Elemente betrachtet und miteinander verglichen werden, da das maximale Element sich nur in der untersten Ebene befinden kann. Es können aber keine weiteren Annahmen zur genauen Positionierung innerhalb der untersten Ebene gemacht werden, weshalb wir hier in jedem Fall eine lineare Laufzeit mit dem (zu vernachlässigen) Faktor $\frac{1}{2}$ erhalten.