Übung 1

1. **Aufgabe**:

   Im Folgenden bezeichne log wie in der Vorlesung den Logarithmus zur Basis 2. Ordnen Sie die 10 Funktionen aus der folgenden Tabelle absteigend nach ihrem asymptotischen Wachstum, d.h., geben Sie eine Anordnung $g_1(n), g_2(n), \ldots, g_{10}(n)$ an, so dass $g_i(n) = \Omega(g_{i+1}(n))$ für $i = 1, \ldots, 9$.

$n \log n$	$(\sqrt{2})^{\log n}$	n^2	$n!$	$4^{\log n}$
$(\frac{3}{2})^n$	n^3	$\log(n!)$	2^{2^n}	$n^{\log\log n}$

2. **Aufgabe**:

   Entscheiden Sie für jede der folgenden Aussagen, ob sie für asymptotisch nicht-negative Funktionen f und g immer wahr, niemals wahr oder manchmal wahr sind:

   (a) $f(n) + g(n) = \Theta(\max(f(n), g(n))$

   (b) $f(n) + O(f(n)) = \Theta(f(n))$

   (c) $f(n) = O(f(n)^2)$

   Begründen Sie jeweils Ihre Antwort.

3. **Aufgabe**:

   Geben Sie ein allgemeines Verfahren an, mit dem man jeden vergleichsbasierten Sortieralgorithmus stabil machen kann.

4. **Aufgabe**:

   Wir betrachten ein spezielles Sortierproblem: Die n zu sortierenden Werte sind als Folge von $\frac{n}{k}$ aufeinanderfolgenden Teilfolgen der Länge jeweils k gegeben. Für jede dieser Teilfolgen gilt, dass ihre Elemente alle größer sind als die Elemente der vorherigen Teilfolge. Geben Sie eine asymptotische untere Schranke für die Anzahl der benötigten Vergleiche für dieses spezielle Sortierproblem an.

5. **Aufgabe**:

   Geben Sie einen Algorithmus an, der eine Folge von n ganzen Zahlen aus dem Bereich $[1..k]$ so vorverarbeitet, dass Anfragen, wie viele Zahlen aus der Folge in einem Anfragebereich $[a..b]$ liegen, $1 \le a \le b \le k$, in Zeit $O(1)$ beantwortet werden können. Ihr Algorithmus sollte Laufzeit $O(n+k)$ erreichen.