21	27	29	35	19	28	19	24
+	+	+	+	+	+	+	+
z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8

Aufgabe 1 [2]

Fügen Sie in obiger Tabelle in den leeren Kästchen, vor denen das Pluszeichen steht, die Ziffern Ihrer Matrikelnummer ein. Führen Sie die Additionen durch und ermitteln Sie die Zahlen z_1 bis z_8.

Aufgabe 2 [18]

a. [9] Schreiben Sie **eine** Funktion in C++ ähnlichem Pseudocode mit einem Parameter n (vom Typ int), deren Laufzeitkomplexität **gleichzeitig** die Ordnungen $O(n^3)$, $\Omega(n)$ und $\Theta(n \log^2 n)$ hat.
b. [9] Fügen Sie in nachfolgender Tabelle Kreuze an den Positionen ein, wo die in der Zeile angeführte Funktion von der in der Spalte angegebenen Ordnung ist.

$f(n)$	$O(n)$	$O(n^3)$	$O(\log n)$	$O(\log^2 n)$	$\Omega(n^2)$	$\Omega(\log n)$	$\Omega(\log^3 n)$	$\Theta(n^2)$	$\Theta(\log n)$	$\Theta(\log^3 n)$
n										
n^2										
n^3										
$\log n$										
$\log^2 n$										
$\log^3 n$										

Anmerkung: $\log^2 n = (\log n)(\log n)$

Aufgabe 3 [20]

Die Werte z_1 bis z_8 (aus Aufgabe 1) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

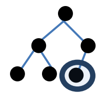
a. [8] Quicksort
b. [4] Bubblesort
c. [8] Counting Sort (**Achtung**: Verwenden Sie für diese Aufgabe die Werte modulo 10, z. B. z_1%10, z_2%10, usf.)

Geben Sie alle notwendigen Schritte so genau an, dass die Arbeitsweise des Algorithmus klar ersichtlich wird.

Aufgabe 4 [20]

a) [5] Geben Sie in C++-ähnlichem Pseudocode die Definition einer effizienten Datenstruktur für einen Max-Heap an, der ganzzahlige Werte speichert.

b) [5] Geben Sie in C++-ähnlichem Pseudocode eine Methode an, die den Knoten im Heap ermittelt, der beim Löschen mit der Wurzel getauscht wird, und dessen Wert ausgibt. Dieser Knoten ist in der üblichen graphischen Darstellung in der untersten Ebene ganz rechts (siehe Abbildung).



Der „Ersatzknoten“ beim Löschen

c) [5] Geben Sie in C++-ähnlichem Pseudocode eine effiziente Methode an, um das Maximum der im Max-Heap gespeicherten Werte auszugeben.

d) [5] Welche Laufzeitordnungen haben Ihre Methoden aus Punkt b) und Punkt c) bezüglich der im Heap gespeicherten Anzahl n der Elemente?

Aufgabe 5 [20]

a) [10] Fügen Sie die Zahlen $\boldsymbol{z_1}$ bis $\boldsymbol{z_7}$ (aus Beispiel 1) in dieser Reihenfolge in eine ursprünglich leere Hashtabelle der Größe 7 ein. Verwenden Sie die Hashfunktion $\boldsymbol{h(x) = x\%7}$ und zur Kollisionsbehandlung double Hashing mit $\mathrm{g(x) = x\%5 + 1}$ als zweiter Hashfunktion. Geben Sie den Zustand der Hashtabelle nach jeder Einfügeoperation an.

b) [4] Geben Sie den Kollisionspfad an, der durchsucht wird, wenn versucht wird, in der nach a) befüllten Hashtabelle zusätzlich $\boldsymbol{z_8}$ (aus Beispiel 1) einzufügen.

c) [2] Wozu wird beim Double Hashing die Markierung „wiederfrei“ verwendet?

d) [2] Warum ist es empfehlenswert, für Double Hashing eine Tabellengröße zu verwenden, die eine Primzahl ist?

e) [2] Nennen Sie 2 dynamische Hashverfahren (Namen reichen aus).

Aufgabe 6 [20]

Gegeben ist die folgende Adjazenzmatrix mit Wegekosten für einen gerichteten Graphen
(die Werte $\boldsymbol{z_1}$ bis $\boldsymbol{z_8}$. sind aus Aufgabe 1 zu übernehmen):

$$\begin{pmatrix} 0 & 0 & z_8 & z_7 & z_6 \\ z_5 & 0 & z_4 & z_3 & 0 \\ z_2 & z_1 & 0 & z_8 & z_7 \\ 0 & 0 & z_6 & 0 & z_5 \\ z_4 & 0 & z_3 & z_2 & 0 \end{pmatrix}$$

a. [2] Skizzieren Sie den gerichteten Graphen.

b. [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die jeweils kürzesten Wege vom Knoten 1 (erste Zeile, erste Spalte der Matrix) zu allen anderen Knoten des Graphen.

c. [8] Bestimmen Sie mit dem Algorithmus von Kruskal einen minimal spannenden Baum des Schattens des Graphen. (Sie erhalten den Schatten des Graphen, indem Sie die Richtungen der Kanten vernachlässigen. Werden dann zwei Knoten durch zwei Kanten verbunden, so werden diese Kanten zu einer zusammengefasst. Anders ausgedrückt: Zwei Knoten x und y im Schatten sind genau dann durch eine ungerichtete Kante verbunden, wenn im ursprünglich gerichteten Graphen zumindest eine der Kanten von x nach y oder von y nach x existiert. Als Gewicht der ungerichteten Kante wählen sie jeweils das Minimum aller durch sie repräsentierten gerichteten Kanten.)