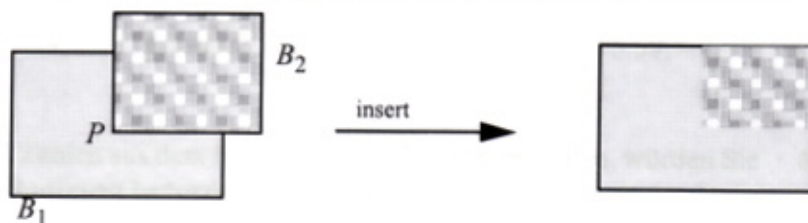


22 Punkte Aufgabe 1 Algebra für Bilder

In dieser Aufgabe soll eine Algebra für Bilder entworfen werden. Jedes Bild hat eine Höhe h sowie eine Breite b (positive ganze Zahlen). Jeder Bildposition $\{(x, y) \mid 1 \leq x \leq b, 1 \leq y \leq h\}$ sei eine Farbe c_{xy} zugeordnet. Die Position $(1, 1)$ bezeichne dabei die linke untere Ecke des Bildes. Jede Farbe sei durch ihr Mischverhältnis von Rot, Grün und Blau im Bereich von 0 bis 1 beschrieben. Sind alle Teile 0, so entsteht die Farbe Schwarz. Weiß erhält man, wenn alle Komponenten auf 1 gesetzt werden. Die Algebra soll folgende Operationen unterstützen:

- *createPicture*: erzeugt ein weißes Bild mit gegebener Größe
- *insert*: fügt ein Bild B_2 in ein anderes Bild B_1 an einer gegebenen Position ein. Dabei werden die überdeckten Bildpunkte des alten Bildes überschrieben. Es werden nur Positionen innerhalb von B_1 berücksichtigt. Die Positionsangabe beschreibt die linke untere Koordinate, an der das neue Bild beginnen soll:



- *rotate*: dreht ein Bild um 90 Grad gegen den Uhrzeigersinn
- *height*: gibt die Höhe eines Bildes zurück
- *width*: gibt die Breite eines Bildes zurück
- *getColor*: gibt die Farbe an einer gegebenen Position zurück
- *setColor*: ändert die Farbe der gegebenen Position
- *mirror*: spiegelt das Bild an der vertikalen Achse in der Mitte des Bildes
- *cut*: erzeugt den Ausschnitt des Bildes, welches durch das achsenparallele Rechteck, das durch zwei Positionen gegeben ist, begrenzt wird.

Hinweise:

Die Funktionen *getColor* und *cut* produzieren jeweils einen undefinierten Wert, wenn die Position bzw. der Bereich außerhalb des Bildes liegt, welches als Argument übergeben wird. Überlegen Sie sich, welche Auswirkungen dies auf die Trägermengen sowie alle anderen Operatoren hat.

Ein zweidimensionales Array der Größe $n \times m$ läßt sich in mathematischer Notation durch eine Folge von $n \cdot m$ Elementen darstellen. Diese können beispielsweise durch ihre Position im Array (i, j) indiziert werden.

- Identifizieren Sie die Sorten der Algebra und geben Sie die Signaturen aller Operationen an.
- Definieren Sie die Trägermengen sowie die Funktionen der Algebra.
- Geben Sie einige Axiome (mind. 5) für die Algebra an, so daß die Operationen *createPicture*, *width*, *height*, *insert*, *getColor*, *setColor*, *rotate* und *mirror* mindestens einmal verwendet werden.

Aufgabe 2 Hashing

12 Punkte

Fügen Sie die aufgeführte Zahlenfolge in eine Hashtabelle mit $m=10$ unter Zuhilfenahme quadratischen Sondierens (Grundform) und der Mittel-Quadrat-Methode als Hashfunktion ein. Geben Sie für jede eingefügte Zahl die Folge der Behälternummern an, die getestet werden. Wie lautet der Inhalt der finalen Hashtabelle?

120 - 163 - 19 - 84 - 69 - 67 - 175 - 59 - 13

Aufgabe 3 Radixsort

14 Punkte

- (a) Es sei *keytype* die Menge aller Zeitstempel der Form *hh:mm:ss* mit $hh \in \{00, 01, \dots, 23\}$ und $mm, ss \in \{00, 01, \dots, 59\}$. Sortieren Sie die unten angegebene Folge mittels Radixsort in drei Phasen.

8 Punkte

23:59:01, 13:31:14, 12:11:01, 03:27:01, 17:23:14, 12:11:03, 17:15:59, 13:28:01

- (b) Wenn Sie 1000 Zahlen aus dem Bereich $1 - 10^{100}$ sortieren sollen, würden Sie Heapsort oder Radixsort bevorzugen? Begründen Sie Ihre Antwort.

6 Punkte

Aufgabe 4 Bestimmung kürzester Wege in Graphen

20 Punkte

Bestimmen Sie die kürzesten Wege vom Knoten *A* zu allen anderen Knoten im unten angegebenen Graphen unter der Verwendung des Algorithmus von Dijkstra. Markieren Sie in jedem Schritt die Menge der gelben und grünen Knoten sowie die roten und gelben Kanten geeignet. Gibt es mehrere Knoten zur Auswahl, wählen Sie den alphabetisch kleineren. Ergeben sich durch das Grünfärben eines Knoten keine Änderungen bei den Kanten, so muß der Graph nicht erneut gezeichnet werden.

