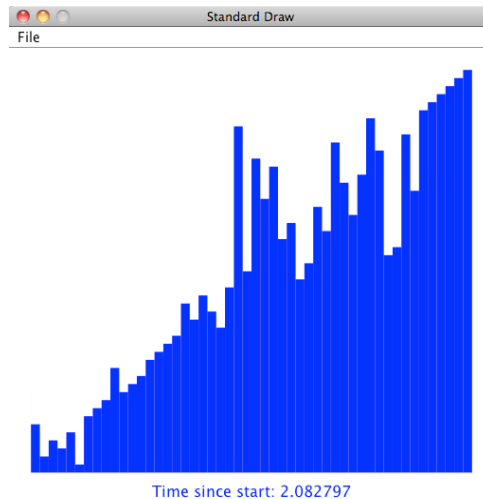


Aufgabenblatt 9 – Animation von Sortierv Verfahren



Klasse BouncingBall

Auf der Web-Seite finden Sie ein kleines Programm, das einen springenden Ball mit Hilfe der Klasse `StdDraw` (siehe Aufgabenblatt 6) animiert. Machen Sie sich damit vertraut.

Klasse ArrayDraw

Die Klasse `ArrayDraw` definiert folgende Methoden:

- `init(int b, int h)` definiert eine weiße Zeichenfläche in der Breite `b` und Höhe `h`
- `draw(int[] a, Color c)` zeichnet das Feld `a` als Balkendiagramm (wie im Bild oben) in der Farbe `c`
- `draw(int[] a, int i, Color c)` zeichnet `a[i]` als einzelnen Balken in der Farbe `c`
- `showTime()` Schreibt die aktuelle Zeit in sec seit Aufruf von `init()` unter das Balkendiagramm. Benutzen Sie dazu `System.nanoTime()`.

Klasse Sort

Die Klasse `Sort` enthält die in der Vorlesung besprochenen Sortierv Verfahren:

- `insertionSort`
- `quickSort`
- `quickSort3Median`

Die Sortierverfahren werden an geeigneten Stellen um Animationsaufrufe erweitert. Wählen Sie folgende Vorgehensweise:

- Nachdem ein Feldelement verschoben wird (z.B. $a[j+1] = a[j]$) oder zwei Elemente vertauscht werden (z.B. $v = a[l]; a[l] = a[r]; a[r] = v;$), wird das gesamte Feld a mit `draw(a, StdDraw.BLUE)` neu gezeichnet.
- Immer wenn ein Feldelement in einem Vergleich beteiligt ist (z.B. $a[j] > v$) wird das entsprechende Feldelement mit `draw(a, i, StdDraw.RED)` übermalt.
- Geben Sie die aktuelle Zeit mit `showTime()` aus.
- Sehen am Ende der beiden `draw`-Methoden eine kleine Verzögerung vor (z.B. `StdDraw.show(10)`).

Klasse SortAnimation:

Schreiben Sie ein Hauptprogramm, das die Animation steuert.

- Das Hauptprogramm erfragt vom Benutzer das Sortierverfahren, das animiert werden soll, die Anzahl der Daten n und ob diese sortiert sein sollen.
- Für eine gefällige Darstellung ist es günstig, ein sortiertes Feld mit den Zahlen von 1 bis n zu generieren und dieses mit einer Funktion `shuffle` gegebenenfalls zu mischen.
- `shuffle` wiederholt n -mal folgenden Zahlfallsschritt: Mit dem Zufallszahlengenerator `Math.random()` werden durch Multiplikation mit $n-1$ zwei zufällige Zahl j und k aus dem Intervall $[0, n-1]$ generiert. Anschließend werden $a[j]$ und $a[k]$ vertauscht.

Abgabe:

Führen Sie Ihr Programm vor und demonstrieren Sie das quadratische Laufzeitverhalten von `insertionSort` und das $O(n \log n)$ -Verhalten von `quickSort` bei zufällig gemischten Daten. Zeigen Sie, dass sich `quickSort` bei sortierten Daten schlecht aber 3-Median-QuickSort gut verhält.