

Übungen zu Algorithmen

Wintersemester 2003/2004

Blatt 5

Aufgabe 5.1 (30 Punkte)

Beim Morsealphabet werden Buchstaben durch kurze und lange Töne kodiert. Diese Nachrichten sollen mit einem endlichen Automaten dekodiert werden.

- Wie lauten die Zustandsmenge und das Eingabealphabet (schriftlich)?
- Entwickeln Sie ein Java-Programm, welches mit einem endlichen Automaten die Umwandlung vornimmt. Beachten Sie dabei fehlerhafte Eingaben und Zustände. Fehlerhafte Zustände liefern als Zeichen den Lattenzaun '#'. Lesen Sie eine Zeile mit Morsezeichen ('.', '-', ',') mit der Methode `readChars()` ein und verarbeiten Sie diese mit Ihrem Automaten.

Das Morsealphabet finden Sie unter `/home/ainf/Uebung/Blatt5/morse.txt`

Hinweis: Ein Zeichen ist komplett, wenn ein Space eingegeben wird. Einzelne Wörter können mit Spaces voneinander getrennt werden.

Zur Erstellung der Übergangsmatrix ist es hilfreich den Zustandsgraphen zu zeichnen.

Beispiel: Der Morsecode

```
.- .-.. --- ---- .. ... - - ---- .-.. .-..
```

liefert als Ergebnis:

```
algo ist toll
```

Aufgabe 5.2 (20 Punkte)

Implementieren Sie in der Klasse `Aufg5_2.java` sechs Methoden, die folgendermaßen aufgerufen werden:

- `int q=quersumme(i);` // berechnet die Quersumme q des übergebenen Integers i ;
- `int c=kgv(i, j);` // berechnet das kleinste gemeinsame Vielfache c der übergebenen Integers i und j ;
- `int f=fakultaet(i);` // berechnet die Fakultät f des übergebenen Integers i ;
- `double m=min(x, y);` // gibt das Minimum m der beiden Doubles x, y zurück;
- `int c=collatz(i);` // gibt den Collatzwert c des Integers i zurück;
- `menue();` // gibt Informationen über die Funktionalität des Programms aus.

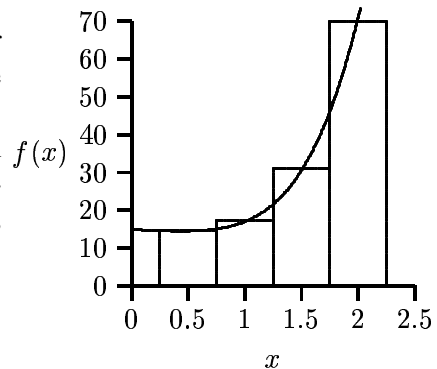
Schreiben Sie in der Methode `main` ein Testprogramm, welches benutzergesteuert die Methoden testet.

Aufgabe 5.3 (30 Punkte)

Schreiben Sie eine Java-Klasse, die das Integral einer Funktion näherungsweise berechnet, indem sie die Fläche unter der Kurve durch Rechtecke annähert (siehe Abbildung). Als Eingaben erwartet das Programm die Grenzen des Integrals (hier: 0.25 und 2.25) und die Zahl der Intervalle (hier: 4). Die Höhe der Rechtecke ergibt sich als Funktionswert in der Mitte der Intervalle.

Schreiben Sie eine Methode für die Funktion:

$$f(x) = x^4 - 10x^2 + x + 50$$



Aufgabe 5.4 (20 Punkte)

Schreiben Sie eine Java-Klasse `Aufg5_4.java`, die eine Zahl $\in [0; 1023]$ einliest und deren 10 Bit-Dualdarstellung ausgibt. Die Klasse soll neben der Methode `main` auch noch die Methoden `istElement` und `bits` enthalten, die folgende Köpfe und Funktionalitäten haben müssen:

```
/** istElement liefert true, wenn zahl in dem geschlossenen
 * Intervall [ugrenze; ogrenze] liegt; sonst false. */
static boolean istElement(int zahl, int ugrenze, int ogrenze)

/** bits liefert die bitanzahl niedrigsten Bits der Dualdarstellung von
 * zahl in der richtigen Reihenfolge als Zeichenkette zurueck.
 * Es muss gelten: 0 <= zahl < 2 hoch bitanzahl */
static char[] bits(long zahl, int bitanzahl)
```

Verwenden Sie zur Implementierung der Methode `bits` den Algorithmus `dezimal → dual` aus dem Skript, Abschnitt 2.4.1.

Bsp.:

Bitte eine Integerzahl aus `[0; 1023]`: 3

Die 10-Bit-Dualdarstellung von 3 lautet: 0000000011

Aufgabe 5.5 (0 Punkte)

Machen Sie sich mit folgenden weiteren UNIX-Kommandos vertraut. Beachten Sie insbesondere die Möglichkeiten zur Ausgabeumlenkung mit `>`, `>>` und `|`.

Kommando	Erklärung
<code>date</code>	Gibt Datum und Zeit aus.
<code>cat datei</code>	Gibt den Inhalt der <code>datei</code> aus.
<code>cat datei more</code>	Gibt den Inhalt der <code>datei</code> seitenweise aus.
<code>date >> datei</code>	Hängt die Ausgabe von <code>date</code> an die <code>datei</code> an.
<code>javac Aufg5_1.java && java Aufg5_1</code>	Nach erfolgreichem Compilieren wird sofort ausgeführt.
<code>javac Aufg5_1.java 2>&1 more</code>	Gibt die Fehlermeldungen des Compilers seitenweise aus.