

Übung zur Vorlesung  
**Einführung in die Programmierung**  
 SoSe 2018 – Blatt 7

Abgabe: Briefkästen RZ/E-Mail bis Montag, den 11.06.2018 um 16:00 Uhr

**Aufgabe 1** (10 Punkte). Ein Struktogramm, auch bekannt unter der Bezeichnung Nassi-Shneiderman-Diagramm, dient der übersichtlichen Darstellung von Algorithmen. Es setzt sich aus den folgenden Elementen zusammen:

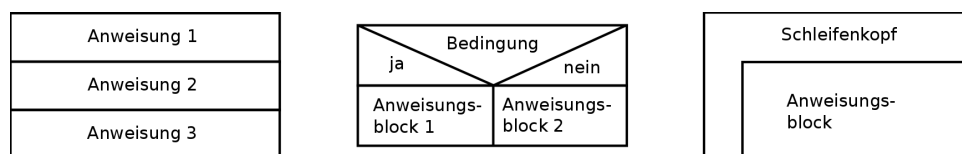


ABBILDUNG 1. Anweisungsblock, Verzweigung und Schleife in einem Struktogramm.

Das Struktogramm eines einfachen Primzahltests sieht etwa wie folgt aus:

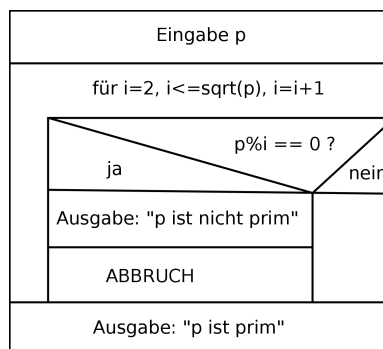


ABBILDUNG 2. Struktogramm eines Primzahltests.

Zeichnen Sie ein Struktogramm eines Algorithmus zur Prüfung der Gültigkeit einer Sudoku-Lösung (vgl. Aufgabe 2, Blatt 5).

**Aufgabe 2** (10 Punkte). Betrachten Sie die beiden folgenden Sortieralgorithmen, welche jeweils eine ungeordnete Liste  $L = [a_1, a_2, \dots, a_n]$  mit  $n$  Einträgen in eine geordnete Reihenfolge bringen. Dabei können die Einträge der Liste beispielsweise Zahlen oder Wörter sein, die in eine aufsteigende bzw. alphabetische Reihenfolge gebracht werden sollen.

**Algorithmus** (Bubblesort). Die Liste  $L = [a_1, a_2, \dots, a_n]$  der Länge  $n > 0$  sei gegeben.

(1) Setze  $i = 2$ .

(2) Für  $k$  von 0 bis  $i - 2$ :

Falls  $a_{i-k} < a_{i-k-1}$ , so vertausche die beiden Einträge.

Falls  $a_{i-k} \geq a_{i-k-1}$  und  $i < n$ , so erhöhe  $i$  um 1 und gehe zu (2).

**Algorithmus** (Mergesort, Spezialfall für  $n = 2^k, k \geq 0$ ). Die Liste  $L = [a_1, a_2, \dots, a_n]$  der Länge  $n = 2^k$  mit  $k \geq 0$  sei gegeben. Der Aufruf des Algorithmus erfolgt mittels  $\tilde{L} = \text{msort}(L)$ , wobei  $\tilde{L}$  die fertig sortierte Liste bezeichnet.

(1) Falls die Länge von  $L$  gleich 1 ist, so ist die Liste bereits sortiert: Setze  $\tilde{L} = L$ .

(2) Falls die Länge von  $L$  gleich  $2^r$  ist mit  $1 \leq r \leq k$ :

Setze  $\tilde{L} = \text{merge}(\text{msort}(L^1), \text{msort}(L^2))$ .

Dabei bezeichnen  $L^1 = [a_1, a_2, \dots, a_{2^{r-1}}]$  und  $L^2 = [a_{2^{r-1}+1}, \dots, a_{2^r}]$  jeweils die linke bzw. rechte Hälfte der Liste  $L$  und der Befehl  $\text{merge}(A, B)$  steht für das Zusammenfügen zweier bereits sortierter Listen  $A, B$  zu einer einzigen sortierten Liste.

Erläutern Sie die Sortieralgorithmen Bubblesort und Mergesort am Beispiel der folgenden Liste von Namen, welche in eine alphabetische Reihenfolge gebracht werden soll (die mathematischen Vergleiche „<“ und „>“ sind in diesem Zusammenhang als lexikographische Vergleiche zu deuten):

[ Hannes, Marie, Jan, Lukas, Josepha, Emanuel, Tatjana, David ]

Sortieren Sie dazu die Liste mit beiden Algorithmen per Hand und schreiben Sie nach jedem Schritt jeweils die aktuelle Reihenfolge der Elemente sowie die Operationen für den darauffolgenden Schritt auf.

**Aufgabe 3** (10 Punkte). Schreiben Sie ein Programm zur Verwaltung eines Telefonverzeichnisses. Definieren Sie dazu einen eigenen Datentyp `struct kontakt`, der sich zur Speicherung von Telefonkontakten eignet. Dieser Datentyp sollte die folgenden Variablen enthalten:

`vorname[20]` und `nachname[20]` vom Typ `char-Array`,

`vorwahl` und `telefonnummer` vom Typ `int`,

`ist_mobil` vom Typ `bool`,

wobei die boolesche Variable `ist_mobil` die Information enthalten soll, ob es sich bei der gespeicherten Telefonnummer um eine Mobilfunknummer handelt. Deklarieren Sie anschließend ein statisches Feld Ihres selbst erstellten Datentyps und geben Sie dem Benutzer die Möglichkeit, einen neuen Eintrag einzulesen und in das Feld zu schreiben oder wahlweise alle bereits gespeicherten Festnetz- oder Mobilfunknummern inklusive der zugehörigen Namen ausgeben zu lassen.

**Aufgabe 4** (10 Punkte). Verwenden Sie die Vektorklasse `std::vector`, um ein einfaches Programm zur Beobachtung der Höchsttemperaturen an verschiedenen Tagen zu schreiben. Deklarieren Sie dazu mittels

```
std::vector<double> temperatur;
```

einen (zunächst leeren) Vektor aus Gleitkommazahlen. Je nach Benutzerwunsch soll dann ein neuer Wert hinzugefügt werden, der zuletzt hinzugefügte Wert wieder gelöscht werden oder wahlweise alle gespeicherten Werte oder die Durchschnittstemperatur aller beobachteten Tage ausgegeben werden. Außerdem sollte der Benutzer die Möglichkeit haben, das Programm zu beenden.

---

Abgabe der Programme per E-Mail, (handschriftlich) kommentierte Ausdrücke der Programme und Rechnungen auf gehefteten, mit Namen versehenen Zetteln in die Briefkästen

Homepage zur Vorlesung: <https://aam.uni-freiburg.de/agba/lehre/ss18/einfprog>