

**Programmierung für Naturwissenschaften
Sommersemester 2015
Übungen zur Vorlesung: Ausgabe am 11.06.2015**

Punkteverteilung: Aufgabe 9.1: 4 Punkte, Aufgabe 9.2: 2 Punkte, Aufgabe 9.2: 4 Punkte

Abgabe bis zum 17.06.2015, 10:00 Uhr.

Aufgabe 9.1 Sei D ein Wörterbuch mit Worten der gleichen Länge. Seien α und ω Worte aus D . Das Wortkettenproblem besteht darin, die kürzeste Folge von Worten und Transformationsschritten von α nach ω zu finden, so dass gilt:

- jeder Transformationsschritt verändert genau einen Buchstaben im transformierten Wort
- jedes Wort, das durch einen Transformationsschritt entsteht, muss in D vorkommen.

Beachten Sie, dass es nicht immer eine Lösung des Wortkettenproblems gibt. Sei $D = \{\text{hit, hot, dot, dog, lot, log, cog}\}$, $d = \text{hit}$ und $d' = \text{cog}$. Dann ist

$\text{hit} \rightarrow \text{hot} \rightarrow \text{dot} \rightarrow \text{dog} \rightarrow \text{cog}$

die kürzeste Folge von Transformationsschritten von d nach d' . Sie hat die Länge 5.

Implementieren Sie ein C++-Programm `wortketten.cpp`, das eine Menge von Worten aus der Datei

`5BuchstabenWoerter` (ein Wort pro Zeile) einliest und für alle Wortpaare in der Datei `Wortpaare.txt` (ein Wortpaar pro Zeile) eine Lösung des Wortkettenproblems für dieses Wortpaar ermittelt.

Das Wortkettenproblem kann man graphentheoretisch lösen. Die Knoten des Graphen sind die Worte aus D und es gibt eine ungerichtete Kante von Knoten u nach v gdw. u und v sich um genau einen Buchstaben unterscheiden. Für jedes Wortpaar (α, ω) muss man im konstruierten Graphen den kürzesten Pfad von α nach ω bestimmen. Dazu kann man eine Breitensuche verwenden.

Implementieren Sie den Graphen, indem Sie die Knoten als Schlüssel w einer STL-map verwenden, deren Wert jeweils die Liste der Worte sind, die mit w adjazent sind (d.h. die sich von w um genau ein Zeichen unterscheiden). Dazu können Sie die folgenden Typsynonyme verwenden:

```
typedef vector<string> Wordlist;  
typedef map<string, Wordlist> Adjacencemap;
```

Sie brauchen nun Funktionen für die folgenden Schritte:

- einlesen der Worte und speichern in einem Container vom Typ `vector<string>`.
- konstruieren der `Adjacencemap`.
- für gegebene Worte α und ω , die im Knoten im Graphen sind muss der kürzeste Pfad berechnet werden. Dazu beginnt man eine Breitensuche bei α . Für jeden Knoten, z.B. v , der das erste Mal besucht wird, z.B. durch eine Kante $u \rightarrow v$, merkt man sich in einem STL-container `previousWord` vom Typ `map<string, string>` den Vorgängerknoten u .

- Nachdem alle Knoten einmal besucht wurden, konstruiert man rückwärts, ausgehend von ω und mit Hilfe von `previousWord` den Pfad zu α und gibt diesen aus.

In den Materialien finden Sie eine Datei mit der erwarteten Lösung für die genannten Eingabedateien. Durch `make test` wird überprüft, ob Ihre Lösung korrekt ist.

Aufgabe 92 In STiNE finden Sie eine Datei `sort_benchmark-c.c`. Schauen Sie sich diese Datei an und implementieren Sie dann ein C++-Programm `sort_benchmark-cpp.cpp` mit der gleichen Funktionalität. Benutzen Sie das mitgelieferte Makefile zum Kompilieren. Verwenden Sie in Ihrer C++-Implementierung den STL-Container `vector` und die Methode `std::sort()`, jedoch nicht die `cstdlib`-Funktion `qsort()`, die im C-code verwendet wurde. Bestimmen Sie die Laufzeit des C-Programms und Ihres C++-Programms. Dazu können Sie die Programmstücke aus dem C-Programm übernehmen, oder sie durch entsprechende C++-Methoden ersetzen.

Welches der beiden Programme ist schneller und um welchen Faktor ist es schneller?

Aufgabe 93 Datei auslesen

Schreiben Sie ein Pythonskript, welches aus der Datei `Messwerte.txt` (STiNE) alle Daten eines Simulationsdurchgangs einliest und jeweils in eine Liste schreibt. Die gesuchten Daten sind: Temperatur (temperature), Druck (pressure), die intermolekulare Energie und die Anzahl der Moleküle außerhalb der Reichweite.

Bestimmen Sie das arithmetische Mittel \bar{x} , die Standardabweichung σ_x und den Wertebereich dieser Daten und geben Sie die Ergebnisse in folgender Tabelle aus:

```
=== ERGEBNISSE ===
- Temperatur: <Mittelwert> +/- <Standardabweichung>
  Wertebereich: <Min> - <Max>
- Druck: <Mittelwert> +/- <Standardabweichung>
  Wertebereich: <Min> - <Max>
- intermolekulare Energie: <Mittelwert> +/- <Standardabweichung>
  Wertebereich: <Min> - <Max>
- Molekuele ausser Reichweite: <Mittelwert> +/- <Standardabweichung>
  Wertebereich: <Min> - <Max>
```

Hinweise:

Das Skript soll keine Interaktion erfordern und auch keine Argumente benötigen, sondern bei Aufruf automatisch die obige Ausgabe erzeugen. (Der Dateiname muss entsprechen statisch im Skript auftauchen, was generell eher nicht üblich ist.)

Mit der Anweisung:

```
if '<suchstring>' in '<text>':
```

kann man z.B. zeilenweise nach Schlüsselwörtern suchen, die dann weitere Anweisungen auslösen (Wert auslesen etc.)

Nutzen Sie folgende Definitionen:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

$$\sigma_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (2)$$

Wobei N die Anzahl der Werte in x ist und x_i der i -te Wert für $1 \leq i \leq N$.

Die Lösungen zu diesen Aufgaben werden am 18.06.2015 besprochen.