



Algorithmen und Datenstrukturen (Informatik II)

SS2001 – Übungsblatt 8

Abgabetermin: 25. Juni 2001

Aufgabe 1. *Vektoren als Klasse, Forts.: operator==, 3 Punkte*

Betrachten Sie:

```
...
    bool operator==(const vektor& w) const;
...
bool vektor::operator==(const vektor& w) const
{
    if ((high-low) != (w.hi()-w.lo())) return false;

    int i_l(low);
    int i_r(w.lo());
    for(; i_l <= high; i_l++, i_r++)
        if ((*this)(i_l) != w(i_r))
            return false;

    return true;
};
```

Testen Sie alle Pfade mit geeigneten Testbeispielen. Schreiben Sie eine Methode `operator!=(const vektor& w) const`, die den Operator `==` benutzt.

Aufgabe 2. *Vektoren als Klasse, Forts.: operator+ und operator*, 7 Punkte*

Schreiben Sie analog Methoden `vektor operator+(const vektor& w) const` und (friend-)Funktionen `friend vektor operator*(const vektor& w, double a)` sowie `friend vektor operator*(double a, const vektor& w)`.

Testen Sie geeignet und dokumentieren Sie Ihre Testplanung und Testdurchführung. Warum ist hier eine doppelte Version von `operator*` erforderlich?

Schreiben Sie eine (friend-)Funktion

`friend double Skalarprodukt(const vektor& v, const vektor& w)`.

Aufgabe 3. *Vektoren als Klasse, Forts.: operator+, 5 Punkte*

Konzipieren und testen Sie friend-Funktionen

```
friend vektor operator+(const vektor& w, double a)
```

und

```
friend vektor operator+(double a, const vektor& w),
```

die die komponentenweise Addition mit einem Skalar realisieren sollen.

Warum ist hier nicht auch „const double d“ bzw. „const double& d“ nötig?

Aufgabe 4. *Vorbereitung: Matrizen als Klasse, 5 Punkte*

Überlegen Sie sich eine Indexabbildung

$$\text{index} : \{RowLow, RowLow + 1, \dots, RowHigh\} \times \{ColLow, ColLow + 1, \dots, ColHigh\} \rightarrow \{0, 1, \dots, (RowHigh - RowLow + 1) \cdot (ColHigh - ColLow + 1) - 1\}$$

die alle Zeilen- und Spaltenindices einer $(RowHigh - RowLow + 1) \times (ColHigh - ColLow + 1)$ -Matrix auf die Indices eines eindimensionalen Vektors mit gleichvielen Elementen bijektiv abbildet.