Einführung in die Programmierung mit C++

Übungsblatt 5

Graph-Matrix Dualität
Graphen

Sebastian Christodoulou Alexander Fleming Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen

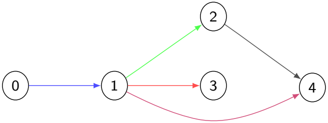
Übung 4
Graph Matrix Dualität

Zuvor (Übungen 2 und 3) hatten wir die Kanten eines Graphs aus einer Textdatei ausgelesen, und in einen `std::vector<int>` kodiert.

`my_graph.txt`

```
5 0 1 2 3 4
5
0 1 1
1 2 1
1 3 1
2 4 1
1 4 1
```

repräsentierter Graph



Vektorkodierung der Kanten

v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	v[6]	v[7]	v[8]	v[9]
0	1	1	2	1	3	2	4	1	4

Diesmal lernen wir eine andere häufige Kodierung: Die Adjazenzmatrixkodierung:

$$
\underbrace{\begin{array}{c|c}
 & \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\end{array}}_{=:M}
$$

Die Adjazenzmatrix eines Graphs ist stets quadratisch. Die Zeilen- und Spaltenanzahl ist durch die Anzahl verschiedener Knoten gegeben (oben: $[0, \ldots, 4]$, also 5 Knoten).

Für alle geraden indizes $i <$ `v.size()` setzt man $M(v[i], v[i+1]) = 1$.

Übung 4
Matrizen als `std::vector` interpretiert

Wir wollen nun mit einer solchen Adjazenzmatrix arbeiten. Da wir momentan nur Vektoren kennen, müssen wir uns damit behelfen, die Matrix in einen Vektor zu kodieren.

Eine Matrix $M \in \mathbb{R}^{p \times q}$ hat genau pq Einträge. Diese Einträge kodieren wir in einem Vektor $w \in \mathbb{R}^{pq}$, indem wir die Zeilen *nebeneinander* schreiben. Die Indizes unserer imaginären Matrix und deren Vektorkodierung w verhalten sich zueinander wie folgt:

$$M(i,j) = w(q * i + j)$$

$$\underbrace{\begin{array}{c c} & \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}}_{=:M} \qquad \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{=:w}$$

Nun wollen wir mit M Matrixmultiplikationen durchführen. Dafür wiederholen wir kurz die Grundlagen

Matrix-Matrix Produkt

Wir gehen vom *Skalarprodukt* auf Vektoren $x, y \in \mathbb{R}^n$ aus. Das Ergebnis ist ein Skalar

$$x^T \; y = \sum_{i=1}^{n} u(i) \cdot y(i)$$

Aus der Vorlesung wissen wir, dass ein Matrix-Vektor Produkt einen Vektor ergibt. Dessen Einträge können als Skalarprodukte aufgefasst werden, wenn wir die Zeilenvektoren von $A \in \mathbb{R}^{m \times n}$ betrachten

$$A \; b = \begin{pmatrix} — & a_1^T & — \\ & \vdots & \\ — & a_m^T & — \end{pmatrix} \begin{pmatrix} | \\ b \\ | \end{pmatrix} = \begin{pmatrix} a_1^T \; b \\ \vdots \\ a_m^T \; b \end{pmatrix}$$

In dieser Übung möchten wir das zu einem Matrix-Matrix Produkt erweitern. Zwei Matrixen $A \in \mathbb{R}^{m \times n}$ und $C \in \mathbb{R}^{n \times q}$ werden multipliziert zu einer Matrix $D \in \mathbb{R}^{m \times q}$, wobei

$$D(i,j) = \sum_{k=1}^{n} A(i,k) \cdot C(k,j)$$

oder anschaulich: Man betrachtet C als Spaltenvektoren, und kann somit die Einträge in D als Skalarprodukte auffassen

$$D = A \; C = \begin{pmatrix} — & a_1^T & — \\ & \vdots & \\ — & a_n^T & — \end{pmatrix} \begin{pmatrix} | & & | \\ c_1 & \dots & c_q \\ | & & | \end{pmatrix} = \begin{pmatrix} a_1^T \; c_1 & \dots & a_1^T \; c_q \\ \vdots & \ddots & \vdots \\ a_n^T \; c_1 & \dots & a_n^T \; c_q \end{pmatrix}$$

Matrix-Matrix Produkt

Einträge der Adjazenzmatrix $M(i,j)$ sind größer als 0 genau dann, wenn es eine Kante von Knoten i zu Knoten j gibt. Wenn wir diese Matrix mit sich selbst multiplizieren, erhalten wir eine neue quadratische Matrix

$$M^2 = MM$$

Die Einträge $M^2(i,j)$ sind größer als 0 genau dann, wenn es einen zwei Schritte langen Pfad von Knoten i zu Knoten j gibt.

Die Adjazenzmatrix M^2 hat genau die gleichen Dimensionen wie M. Wir können M mit sich selbst beliebig oft multiplizieren.

$$M^n = \underbrace{M\,M \cdots M}_{n \text{ mal}}$$

Einträge in der Matrix $M^n(i,j)$ sind genau dann größer als 0, wenn es einen n Schritte langen Pfad von Knoten i zum Knoten j gibt.

1. Berechnen Sie die Adjazenzmatrix für den Graph aus `my_graph.txt`. Diese Matrix soll in einer `std::vector<int>` gespeichert werden.

2. Ergänzen Sie die folgenden Funktionen:

    - `extract_column`: Diese Funktion nimmt eine Matrix in Vektordarstellung und einen Spaltenindex. Sie gibt eine Kopie der entsprechenden Spalte zurück.
    - `extract_row`: Diese Funktion nimmt eine Matrix in Vektordarstellung und einen Zeilenindex. Sie gibt eine Kopie der entsprechenden Zeile zurück.
    - `dot_product`: Diese Funktion nimmt zwei Vektoren, berechnet deren Skalarprodukt und gibt das Ergebnis zurück.
    - `matrix_matrix_product`: Diese funktion nimmt zwei Matrizen und berechnet deren Matrix-Matrix-Produkt
        - Das Matrix-Matrix-Produkt $D = A\ C$ kann als Skalarprodukte der Zeilen A und Spalten von C betrachtet werden (Seite 4).
        - Benutzen Sie die obigen Funktionen zur Berechnung der Einträge von D. Geben sie Matrix D in Vektordarstellung zurück.

3. Für welche Knoten k in dem Graph aus `my_graph` gibt es einen Pfad $k \to k$, der *genau* 3 Schritte lang ist? Geben diese Knoten aus dem Terminal aus.

Abgaben

- main.cpp