

Úloha 13 – Výpočet součinu matic

Zadání

Vytvořte program, který z textového souboru načte matice **A**, **B** a provede jejich součin, je-li definovaný. Výslednou matici $C = A * B$ program uloží do textového souboru.

Rozbor problematiky

Formální definice

Součinem dvou matic **A** a **B**, pro které platí, že rozměr matice **A** je $c \times d$ a rozměr matice **B** je $d \times f$, je matice **C** s rozměry $c \times f$. Její výpočet je dán následujícím vztahem:

$$(A \cdot B)_{ij} = \sum_{p=1}^n a_{ip} \cdot b_{pj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{id}b_{dj}$$

Popis průběhu programu

Základním krokem bylo vytvoření třídy *Matrix*, která reprezentuje matici. Matice je zde reprezentována jako dvoudimensionální neboli 2D list – tedy seznam seznamů, kdy každý jeden seznam odpovídá řádce matice, například:

$$A = \begin{pmatrix} 2 & -3 & 7 \\ 1 & 2 & 2 \end{pmatrix}$$

matice bude pro potřeby výpočtu zapsána v reprezentaci jako:

```
[[ '2', '-3', '7'], [ '1', '2', '2']]
```

a následně rozdělená na řádky:

```
0: [ '2', '-3', '7']
```

```
1: [ '1', '2', '2']
```

Program nejprve vyzve uživatele k zadání názvu souboru, který obsahuje matici ve správném tvaru – jedná se o textové soubory s vypsányými čísly v řádkách a sloupcích ve formátu matice. Po zadání názvů souborů program validuje, zda soubor existuje, zda má k němu přístup a zda jsou data ve správném formátu. Ve chvíli, kdy je vše v pořádku zadáno program inicializuje matice 1, 2 (v našem případě *matrix1.txt* a *matrix2.txt*) a rozdělí ho na řádky. Jakmile je

rozděleno na řádky, odešle se část do funkce *parse* a v tam se podle mezer rozdělí řádky na sloupce. Zde se nastaví, že šířka budoucí matice je délka řádku matice původní, v případě, že zadaný soubor má špatný tvar – tudíž není matice, program zahlásí chybu a vypne se. Pokud projde, do reprezentace se zapíše celý řádek. Funkce se opakuje pro každý řádek.

Když máme soubory přečtené a validní, můžeme zavolat funkci *matrix_multiply*. Nejprve se ověří, zda se šířka matice 1 rovná matici 2 (pokud ne, program oznámí a vypne se). Inicializujeme *result*, kdy výsledná matice bude mít šířku jako matice 2 a výšku jako matice 1. Když projde v pořádku, tak ve třídě *Matrix*, jež nebude platit, že šířka a výška se rovná nule, a tudíž proběhne reprezentace, která připraví výšku a šířku výsledné matice s prázdnými hodnotami, aby se nestalo, že program sáhne mimo pole. Následné for cykly prochází indexy výšky matice 1 a šířky matice 2 po řádcích i sloupcích, které pak vychází z obecného pravidla pro součin matic. Výsledek se vrátí jako *result* a do funkce *save*, která celý výsledek uloží. Funkce projde všechny řádky a číslo po čísle ukládá do předem definovaného tvaru matice.

Popis funkcí

Hlavní částí celého programu je třída *Matrix*, která obsahuje následující funkce:

```
def __init__(self, width=0, height=0)
```

Funkce na začátku přímá matici 1 a matici 2 (prázdné), takže *width* a *height* konstruktor mu nepředáme, provede `else – self.representation = []`, utvoří se tedy prázdná reprezentace, kde *self.width* a *self.height* je 0.

V případě, kdy matice mají rozměr (*width*, *height* > 0) provede se *self.representation*, která připraví pole o velikosti výsledné matice.

```
def read(self, input_name)
```

Čtecí funkce, která rozdělí soubor na řádky a pošle ho do funkce *parse*. Když soubor neodpovídá požadavkům na vstupu, validuje jeho existenci, přístup k němu a zda se celkově jedná o validní soubor. V případě chyby vše a nahlásí a vypne se.

```
def parse(self, lines)
```

Na parametru přímá *self* a *linie*, následně rozdělí matici podle mezer, nastaví, že šířka matice je délka budoucí matice. V případě, že matice nemá správný tvar, vyskočí hláška (např. v jednom z řádků chybí jedno číslo – tudíž nejde o matici). Pokud projde, uloží do *self.representation* celý řádek a výsledné výšce přidá jedničku. Opakuje se pro každý řádek.

```
def save(self, output_file_name)
```

Funkce projde celou maticí a ukládá řádky, nejprve uloží číslo, mezeru a následně číslo další, funkce pozná, že jde o poslední řádek, a tak již nepřipravuje řádek nový pro další zápis. V případě, že nejde soubor uložit kvůli oprávnění, nebo kvůli jinému problému nemohl program výsledný textový soubor uložit, program nahlásí chybovou hlášku.

```
def matrix_multiply(matrix1, matrix2)
```

První, co funkce ověří je, jestli jde o matice, kde je definované jejich násobení, v případě, že není, vytiskne hlášku a ukončí aplikaci. Aplikace třemi vnořenými for cykly postupně prochází indexy. Nejprve index výšky matice 1, pak index šířky matice 2 a sloupec matice 2. Výsledná reprezentace výsledku se již řídí obecným pravidlem pro výpočet součinu matic.

Vstupy a výstupy

Vstup

Vstupem pro program jsou dvě libovolné matice napsané ve správném tvaru do textového souboru. Soubory jsou pojmenovány matrix1.txt a matrix2.txt. V textovém souboru nesmí být uložen žádný chybný znak např. mezera či řádek navíc. Program v tomto případě vypíše chybovou hlášku a vypne se.

Výstup

Výstup definovaného součinu dvou matic je vygenerován textový soubor obsahující výsledný součin matic A, B pojmenovaný output.txt obsahující matici C.