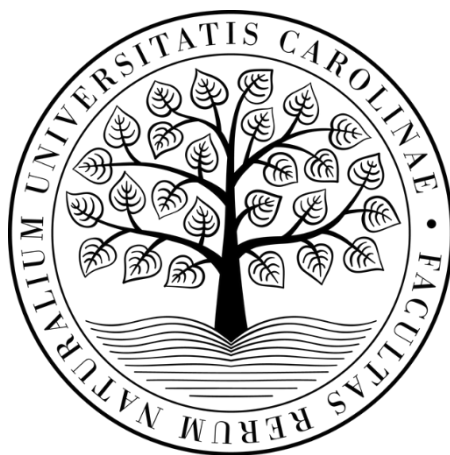


Univerzita Karlova

Přírodovědecká fakulta



Výpočet součinu matic

Úvod do programování

Miroslav Hruběš

3. BGEKA

Světice, 2022

Zadání

Vytvořte program, který bude počítat součin dvou matic. Součástí úlohy bude kromě aplikace také dokumentace, o rozsahu 4-5 stran ve formátu PDF, která bude obsahovat následující:

- rozbor problému
- existující algoritmy
- popis zvoleného algoritmu
- strukturu programu (datové struktury, metody,...)
- popis vstupních/výstupních dat
- problematická místa
- možná vylepšení

Program bude považována za nefunkční pokud:

- při zpracování dat dojde k pádu (runtime chyby,...)
- vrací špatné výsledky
- neřeší možné singulární případy

Rozbor problému

Matice je uspořádání čísel obdélníkového či čtvercového tvaru. Mezi standardní operace s maticemi patří i vynásobení dvou matic. Obecná definice součinu matic vypadá takto:

Nechť existuje matice A o rozměrech $m \times n$ a matice B o rozměrech $n \times p$. Jejich součin $A \cdot B$ je matice o rozměrech $m \times p$. Každý prvek matice součinu je definován vztahem:

$$(A \cdot B)_{ij} = \sum_{r=1}^n a_{ir}b_{rj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots a_{in}b_{nj}$$

Z definice jasně vyplývá podmínka, že počet sloupců první matice se musí rovnat počtu řádků druhé matice. Déle je nutné uvést, že násobení matic je nekomutativní, tzn.:

$$A \cdot B \neq B \cdot A$$

Je tedy velice důležité si před samotným násobením matic uvědomit, kterou maticí se kterou chceme násobit.

Existující algoritmy

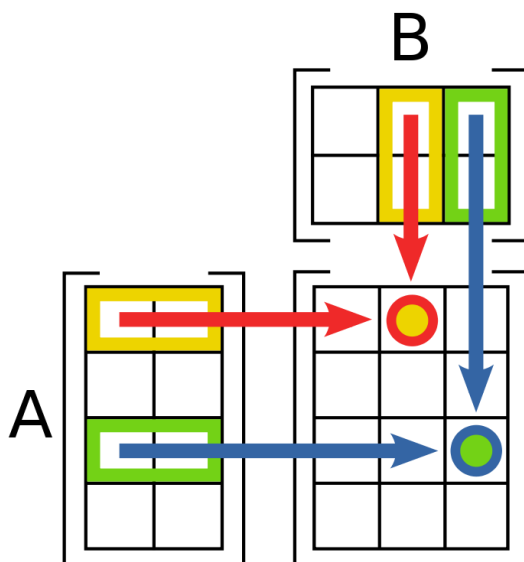
Základní a nejjednodušší algoritmus je iterační algoritmus známý také jako klasické násobení. Ten v sobě zahrnuje cyklus v cyklu, kde podle aktuálních indexů prvku, které se s každou iterací mění, probíhají výpočty prvků nové matice. Tento algoritmus se stal základem pro tvorbu programu. Jeho jednoduchost je ovšem vykoupena časovým vytížením. Pro čtvercové matice $n \times n$ proběhne n^3 výpočtů.

Kvůli časové náročnosti výpočtů byl hledán algoritmus, který bude rychlejší. Jedním takovýmto algoritmem je Strassenův algoritmus, pojmenovaný podle německého matematika Volкера Strasse. Svůj algoritmus popsal v roce 1969. Na rozdíl od klasického násobení, které při maticích o rozměrech 2×2 potřebuje 8 násobení a 4 sčítání, Strassenův algoritmus potřebuje 7 násobení a 18 sčítání. Jelikož je násobení náročnější operace než sčítání, je tento algoritmus o trochu rychlejší. Ještě rychlejší je tento algoritmus v tzv. Winogradově úpravě. Ta vyžaduje stále 7 násobení, ale už jen 15 sčítání. Časová náročnost pro matice $n \times n$ je zde přibližně $n^{2,807}$ výpočtů. Strassenův algoritmus má ale také i nevýhody. Kvůli větší složitosti je potřeba provést více pomocných operací. Také má tento algoritmus díky své rekurzivní podobě výrazně větší nároky na paměť.

Dalším rychlejším algoritmem je Coppersmithův-Winogradův algoritmus, který je pojmenovaný po Donovi Coppersmithovi a Shmuelovi Winogradovi. Algoritmus a jeho následná vylepšení jsou zatím nejrychlejšími známými algoritmy na součin matic. V maticích $n \times n$ provede algoritmus $n^{2,375}$ výpočtů. Algoritmus se často používá jako stavební blok v jiných algoritmech k prokázání teoretických časových hranic. Ovšem na rozdíl od Strassenova algoritmu se v praxi na součin matic příliš nepoužívá, protože je výhodný pouze pro matice takových rozměrů, že nejdou zpracovat moderními hardwary.

Popis zvoleného algoritmu

Jak už bylo zmíněno, pro tvorbu programu byl vybrán iterační algoritmus známý také jako klasické násobení. Každý prvek nově vzniklé matice součinu je zde definován jako součet součinů prvků obou matic i -tého řádku první matice a j -tého sloupce druhé matice. Znázornění této operace je na Obr. 1. Pro matice o rozměrech $n \times n$, je časová náročnost n^3 výpočtů.



Obr. 1: Součin matic

Pseudokód tohoto algoritmu v programu je následující:

Vytvoř prázdný seznam ,mat‘.

Tolikrát, kolik je počet prvků první matice,

vytvoř prázdný seznam ,o‘.

pro proměnnou ,j‘, která začíná na 0 a postupně se o 1 zvyšuje, než dosáhne počtu prvků nultého prvku druhé matice,

přidej do seznamu ,o‘ prvek 0.

přidej seznam ,o‘ jako prvek do seznamu ,mat‘.

Tolikrát, kolik je počet prvků první matice,

pro proměnnou ,j‘, která začíná na 0 a postupně se o 1 zvyšuje, než dosáhne počtu prvků nultého prvku druhé matice,

pro proměnnou ,k‘, která začíná na 0 a postupně se o 1 zvyšuje, než dosáhne počtu prvků druhé matice,

přičti k j-tému prvku z i-tého prvku seznamu ,mat‘ součin k-tého prvku z i-tého prvku první matice a j-tého prvku z k-tého prvku druhé matice.

Struktura programu

Vstupní data jsou na počátku programu zadána uživatelem prostřednictvím zavolání metody na to uzpůsobené. Uživatel nejprve zadá počty sloupců a řádků obou matic. Pokud uživatel nezadá počet řádků druhé matice stejný jako počet sloupců první matice, což je podmínka u matic nutná k provedení operace součinu, program tuto skutečnost uživateli oznámí a program ukončí. Oznámení a ukončení programu se provede také v případě, že uživatel nezadá do počtu řádků a sloupců celé číslo.

Uživatel následně opět prostřednictvím zavolání metody na to uzpůsobené zadá hodnoty prvků obou matic. Hodnoty jsou zadávány vždy pro jeden řádek. Pro zadání dalšího prvku je nutné nejprve zmáčknout klávesu Enter. V případě, že uživatel nezadá do hodnot prvků matice číslo, program tuto skutečnost uživateli oznámí a program ukončí. Hodnoty se ukládají do seznamu a po naplnění daného řádku je seznam přidán do seznamu dané matice. Proměnná „matice“ je tedy seznam seznamů, kde každý seznam ze seznamu představuje prvky jednoho řádku matice. Program uživateli vždy oznámí, pro který řádek které matice právě zadává hodnoty. Jako problémové by se mohlo jevit zadávání čísla ve formě zlomku (např. $\frac{1}{4}$). Tato skutečnost je ovšem v programu ošetřena. Vstupní hodnota je nejprve převedena na proměnnou typu *string* a poté vložena do cyklu, který zjišťuje, zda se ve vstupní hodnotě vyskytuje znak lomítka. Pokud ano, znaky před a po lomítku jsou převedeny na proměnnou typu *float* a následně dány do podílu. Pokud program lomítko nenajde, program provede prosté převedení vstupu na proměnnou typu *float*.

Po naplnění obou matic prvky dojde prostřednictvím další metody k výpočtu součinu obou matic. Tento výpočet vysvětluje pseudokód napsaný v předchozí kapitole. Výsledkem je proměnná „matice“, tedy seznam seznamů, kde každý seznam představuje řádek nově vzniklé

Posledním krokem programu je vytisknutí všech tří matic do terminálu. Zavolaná metoda postupně vytiskne všechny prvky jedno řádku matice do jednoho řádku v terminálu. Každý řádek v terminálu začíná a končí hranatými závorkami.

Případné vylepšení programu by mohlo být zavedení externího vstupu a externího výstupu programu, tzn., že by na počátku programu vstupní hodnoty nezadával uživatel, ale program by vzal konkrétní soubor, resp. soubory (např. formátu txt), ve kterém by byly vstupní hodnoty již uvedeny. Stejně tak by mohla být výstupní matice vytištěna do externího souboru místo do konzole. V tomto případě by bylo ale potřeba věnovat více pozornosti převedení

vstupních dat z řetězců textového formátu do proměnných typu *float* a to včetně speciálních případů, jako jsou čísla ve formě zlomku.

Zdroje

https://cs.wikipedia.org/wiki/N%C3%A1soben%C3%AD_matic

<https://people.fjfi.cvut.cz/pelandedi/tema/SkriptaTEMAKonecSemestru/StrassenZvonik.pdf>

https://en.wikipedia.org/wiki/Matrix_multiplication_algorithm#Divide-and-conquer_algorithm

https://handwiki.org/wiki/Coppersmith%E2%80%93Winograd_algorithm