Univerzita Karlova

Přírodovědecká fakulta

Výpočet součinu matic

Úvod do programování

Miroslav Hrubeš

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 × n a matice B o rozměrech n × p. Jejich součin A · B je matice o rozměrech m × p. Každý prvek matice součinu je definovaná vztahem:

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

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 matici se kterou chceme násobit.

Existující algoritmy

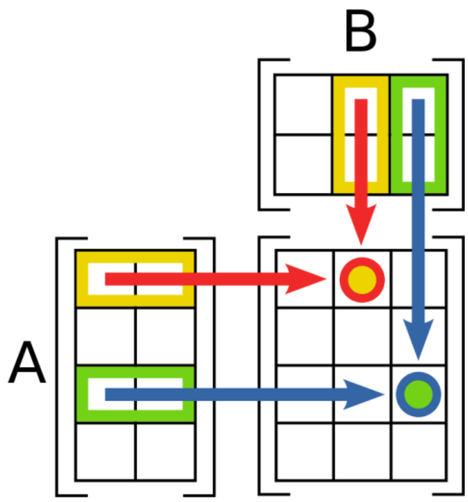
Základní a nejjednodušší algoritmus je iterační algoritmus známý také jako klasické násobení. Ten v sobě zahrnuje cyklus v cyklu, kde podle aktuálních indexů prvku, které se s každou iterací mění, probíhají výpočty prvků nové matice. Tento algoritmus se stal základem pro tvorbu programu. Jeho jednoduchost 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 Volkera 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 nejrychlejším 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.



Obr. 1: Součin matic

Algoritmus tvoří dva v sobě vnořené cykly. Ve vnitřním cyklu dochází k výpočtu součinu prvků i-tého řádku první matice a j-tého sloupce druhé matice. Součiny se poté vzájemně sečtou a výsledná hodnota je prvek matice součinu na pozici i-tého řádku a j-tého sloupce. Pro matice o rozměrech $n \times n$, je časová náročnost n^3 výpočtů.

Struktura programu

Vstupní data jsou na počátku programu zadána uživatelem. 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 zadá hodnoty prvků obou matic. Hodnoty jsou zadávány vždy pro jeden řádek. Pro zadání dalšího prvku je nutné nejprve zmáčknout klávesu Enter. 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. ¼). 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 k výpočtu součinu obou matic. Nejprve je z každého řádku druhé matice vybrán prvek na první pozici a uložen do pomocného seznamu. Po naplnění tohoto pomocného seznamu je každý prvek tohoto pomocného seznamu vynásoben s každým prvkem prvního řádku první matice. Všechny tyto násobky jsou postupně ukládány do proměnné, která je po konci výpočtů uložena do seznamu prvků výsledné matice. Zároveň po konci prvního násobení dojde následně k násobení prvky pomocného seznamu s prvky druhého řádku první matice. Proces se opakuje do okamžiku, kdy jsou vytvořeny všechny prvky prvního sloupce výsledné matice. V tu chvíli se pomocný seznam vyprázdní a naplní se prvky z druhého sloupce druhé matice. Celý proces se dále opakuje. Výsledkem je tedy seznam, o počtu pozic rovnému součinu řádků a sloupců nově vzniklé matice, který je vyplněn prvky nově vzniklé matice.

Aby byl tento seznam přeměněn na proměnnou „matice“, tedy seznam seznamů, kde každý seznam představuje řádek nově vzniklé matice, je potřeba výsledný seznam přeindexovat. Seznam prvků matice součinu, prochází cyklem, ve kterém se program ptá, zda je jeho pozice v tomto seznamu po vydělení počtem řádků první matice rovna proměnné ‚citac’, který má nastavenou hodnotu 0. Prvky, které splňují tuto podmínku, jsou předány do pomocného seznamu a po konci tohoto cyklu je tento seznam přidán do proměnné výsledné matice. Zároveň dojde k navýšení hodnoty proměnné ‚citac‘ o 1. Proces se opakuje a dochází k naplnění pomocného seznamu prvky dalšího řádku nově vzniklé matice.

Posledním krokem programu je vytisknutí všech tří matic do terminálu. Opět je použit cyklus, který postupně vytiskne všechny prvky jedno řádku matice do jednoho řádku v terminálu. Každý řádek v terminálu začíná a končí hranatými závorkami.

Jako případné vylepšení programu by se dala uvažovat modifikace, která by se ve výpočtu součinu neřídila podle sloupců druhé matice, ale podle řádků první matice. V tomto případě by nebylo potřeba přeindexování prvků výsledné matice a nově vzniklý řádek nově vzniklé matice by se mohl hned vytisknout do terminálu. Další vylepšení by mohlo být v podobě 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/pelanedi/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