Zadání třetího domácího úkolu

Teorie

Gaussovu eliminační metodu (GEM) jistě dobře znáte z lineární algebry. Slouží k mnoha užitečným věcem, pro nás bude nejdůležitější řešení soustav lineárních rovnic a hledání inverzní matice.

Zadání

Vytvořte funkci `GEM(M, solveSystem = F, inverse = F)`, kde `M` je matice. Pomocí Gaussovy eliminační metody ji převedete do tvaru, kde vlevo je diagonální matice a vpravo "zbytek". V případě, že má matice lineárně závislé řádky bude výsledná matice obsahovat dole blok 0. Znamená to, že Gaussovu eliminaci provedete "dvakrát"- jednou "dolů"abyste získali horní trojúhelníkovou, podruhé "nahoru"abyste získali diagonální (a to s jedničkami na diagonále).

V případě že `solveSystem = F` a `inverse = F`, program vrátí a vypíše výsledek této Gaussovy eliminace.

V případě, že `solveSystem = T` a `inverse = F` interpretujte matici jako rozšířenou matici systému lineárních rovnic. Funkce pak vrátí a vypíše vektor který je řešením této soustavy, pokud řešení existuje a je jednoznačné. Pokud neexistuje, vrátí `None` a vypíše, že systém nemá řešení. Pokud je řešení nejednoznačné, vypíše tuto informaci a vrátí opět `None`. V případě, že `solveSystem = F` a `inverse = T` spojte matici `M` s jednotkovou maticí a proveďte opět GEM. Po úpravách bude na místě jednotkové matice inverzní matice (pokud má `M` inverzi). V takovém případě tuto inverzní matici vraťte a vypište. Pokud `M` inverzní matici nemá, tuto informaci vypište a vraťte `None`. V případě, že `solveSystem = F` a `inverse = T` program ohlásí, že tato kombinace není možná a vrátí `None`.

Podmínky

Na řešení jsou kladeny tyto podmínky:

- Jedná se o nejkomplexnější úkol, na jeho vypracování máte 3 týdny. Deadline pro odevzdání je **9. 5. 23:59:59**.
- Během této doby můžete využívat konzultace, a to i individuálně po domluvě.
- Pokud úkol odevzdáte do 3. 5. (včetně), napište mi email, já úkol co nejrychleji opravím a budete mít možnost ještě mé připomínky zapracovat.
- Kód jste vytvořili sami.
- Kód je čitelný, komentáře i názvy proměnných jsou anglicky.
- Kód nepoužívá žádné externí moduly.
- Ke kódu jsou přiloženy všechny funkce, které jsou potřeba.
- Kód dělá, to co je požadováno v zadání
- Na konci demonstrujete funkčnost kódu na příkladech níže.
- Můžete používat libovolné funkce, ale samotný výpočet systému řešení resp. inverzní matice musí proběhnout pomocí GEM, ne pomocí Cramerova pravidla nebo Laplaceova rozvoje. Můžete ale například pomocí determinantu ověřit, zda k matici existuje inverze.
- Je lepší odevzdat vlastní kód, který neřeší korektně všechny možné situace než řešení opsat. Víc se toho naučíte.

Hints

Tento úkol se může zdát děsivým a komplikovaným. Je důležité se nezaleknout a začít s jednoduchým funkčním prototypem, který budete dále rozšiřovat. Zde je postup, který bych volil já (můžete, ale nemusíte se jej držet).

- Předpokládejte, že je matice hezká a naimplementujte první část GEM - tvorbu horní trojúhelníkové matice.
- Vyřešte případné problémy - lineárně závislé řádky nebo sloupce. Je lepší je detekovat v průběhu GEM nebo ještě před tím?
- Vyřešte gaussovu eliminaci "nahoru"tak, aby se vytvořila v levém horním bloku jednotková matice. Teď je skoro hotovo.
- Vyřešte, jak pro `solveSystem = T` resp. `inverse = T` z výsledku extrahovat požadovanou část a tu vrátit (vypsat).
- Snažte se dosáhnout toho, aby váš kód korektně zpracoval testovací vstupy níže. To je požadovaná laťka.
- Možná je to neintuitivní, ale někdy je lépe začít s "větším"vstupem. Na testovací matici 2×2 se většina chyb ve vašem kódu neprojeví. Na matici 5×5 už spíš ano.

Testovací vstupy

```
A = [[2,   1, -1,   8],
     [-3, -1,  2, -11],
     [-2,  1,  2,  -3]]

GEM(A)
>>> [[1.0, 0.0, 0.0, 2.0], [0.0, 1.0, 0.0, 3.0], [-0.0, -0.0, 1.0, -1.0]]

GEM(A, solveSystem = T)
>>> [2.0, 3.0, -1.0]

GEM(A, inverse = T)
>>> This matrix is singular.

B = [[1, 3, 5],
     [1, 1, 1],
     [1, 2, 4]]

GEM(B)
>>> [[1.0, 0.0, 0.0], [-0.0, 1.0, 0.0], [0.0, 0.0, 1.0]]

GEM(B, solveSystem = T)
>>> This system has no solution.

GEM(B, inverse = T)
>>> [[-1.0, 1.0, 1.0], [1.5, 0.5, -2.0], [-0.5, -0.5, 1.0]]

C = [[1, 2],
     [1, 1],
     [1, 3]]

GEM(C)
>>> [[1.0, 0.0], [0.0, 1.0], [0.0, 0.0]]

GEM(C, solveSystem = T)
>>> This system has no solution.

GEM(C, inverse = T)
>>> This matrix is singular.
```