

Odwrotność macierzy wymiaru większego niż 3x3

Wiemy już w jaki sposób obliczać macierz odwrotną dla wymiarów 2x2 oraz 3x3. Ale co jeżeli chcielibyśmy obliczyć odwrotność macierzy 5x5?

Istnieją dwie metody obliczania macierzy odwrotnej:

- Metoda wyznacznikowa – czyli taka, do wyznaczenia której potrzebny nam jest wyznacznik.
- Metoda bez wyznacznikową- korzystanie z zależności macierzy jednostkowej.

O ile przy szukaniu macierzy odwrotnej 2x2 i 3x3 metoda wyznacznikowa nie przysparzała nam większych kłopotów, tak przy macierzach wymiaru większego niż 3x3 samo szukanie wyznacznika może być bardzo pracochłonne.

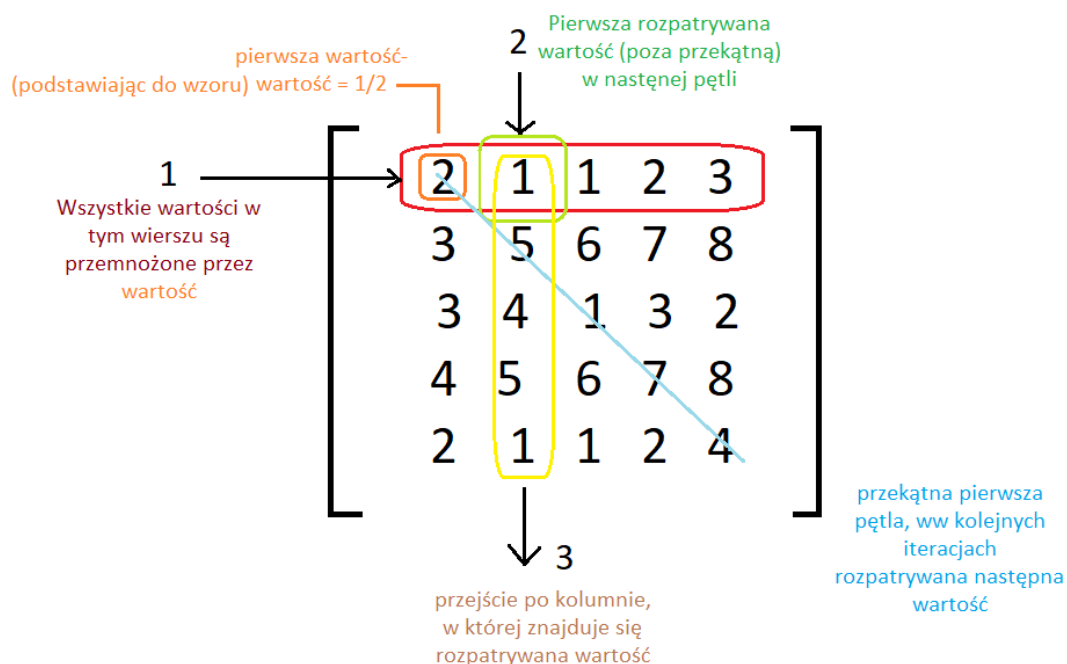
Dlatego przy wyznaczaniu większych macierzy odwrotnych posłużymy się metodą bez wyznacznikową Gaussa-Jordana.

Aby obliczyć naszą macierz odwrotną posłużymy się wzorem :

$$A * X = I * B$$

$$I * X = A^{-1} * B$$

Metoda Gaussa-Jordana jest jedną z dwóch metod rozwiązywania układów równań przy pomocy operacji elementarnych na macierzach. W metodzie tej sprowadzamy macierz rozszerzoną układu równań do postaci bazowej (macierzy jednostkowej). Z tej postaci odczytujemy wprost rozwiązania układu równań.



Do zastosowania powyższej metody potrzebne nam będą:

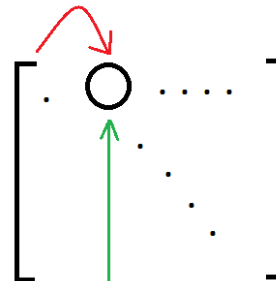
- Macierz jednostkowa
- Funkcja na mnożenie macierzy
- Funkcja na kopiowanie macierzy

Poszczególne kroki implementacji:

1. Tworzymy macierz główną – tą którą chcemy odwrócić
2. Tworzymy macierz jednostkową, o takich samych wymiarach jak macierz główna
3. Zapisujemy kopie obu macierzy (pracujemy na kopiach)
4. Pobieramy długość macierzy głównej
5. Uruchamiamy pętlę przechodzącą po długości macierzy głównej (wiersze):
 - Przeliczamy wartości po przekątnej macierzy głównej:
 $\text{Wartość} = 1 / \text{konkretna_wartość_przekątnej}$
 - Uruchamiamy kolejną pętlę przechodzącą po długości macierzy głównej(kolumny):
 - **Wartość**(pozyskaną w konkretnej iteracji) mnożymy przez wszystkie kolumny w danym wierszu macierzy głównej jak i jednostkowej (czyli wartość przekątnej, która mamy aktualnie – przemnażamy wszystkie kolumny z wiersza, w którym leży)
 - Tworzymy listę indeksów, które znajdują się w naszej macierzy głównej
 - Uruchamiamy pętlę, która będzie przechodziła przez indeksy macierzy, omijając wartości umieszczone na przekątnej(wiersze):
 - W trakcie trwania pętli tworzymy **zmienną**, która będzie przechowywała tymczasowo wartości macierzy głównej aktualnie rozpatrywane w iteracji
 - Inicjujemy kolejną pętlę przechodzącą przez długość macierzy głównej(kolumny):
 - Zmieniamy wartości w macierzach (głównej i jednostkowej) oprócz przekątnej

Od wartości bazowej odejmujemy **zmienną** pomnożoną przez wartość, której indeksy to [numer iteracji pętli głównej][numer kolumny z aktualnej pętli]

$\text{wartość}[0][1] - \text{zmienna} * \text{wartość}[0][0]$



rozpatrujemy tę wartość macierzy
załóżmy, że jej pozycja to [0][1]

6. Zmodyfikowana macierz jednostkowa, jest naszą macierzą odwrotną, natomiast zmodyfikowana macierz główna powinna mieć formę macierzy jednostkowej.

Poniżej przedstawiam przykładowe działanie algorytmu na macierzy 5x5

```
1  Macierze startowe to:
2  AM Matrix IM Matrix
3  [' +5.000', ' +4.000', ' +3.000', ' +2.000', ' +1.000'] [' +1.000', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
4  [' +4.000', ' +3.000', ' +2.000', ' +1.000', ' +5.000'] [' +0.000', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
5  [' +3.000', ' +2.000', ' +9.000', ' +5.000', ' +4.000'] [' +0.000', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
6  [' +2.000', ' +1.000', ' +5.000', ' +4.000', ' +3.000'] [' +0.000', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
7  [' +1.000', ' +2.000', ' +3.000', ' +4.000', ' +5.000'] [' +0.000', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
8
9
10 Korzystając z powyższych macierzy, przeskaluj rząd-1 AM i IM według elementu diagonalnego 1 AM, który wynosi 1/+5.000.
11
12 AM Matrix IM Matrix
13 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
14 [' +4.000', ' +3.000', ' +2.000', ' +1.000', ' +5.000'] [' +0.000', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
15 [' +3.000', ' +2.000', ' +9.000', ' +5.000', ' +4.000'] [' +0.000', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
16 [' +2.000', ' +1.000', ' +5.000', ' +4.000', ' +3.000'] [' +0.000', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
17 [' +1.000', ' +2.000', ' +3.000', ' +4.000', ' +5.000'] [' +0.000', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
18
19 Korzystając z powyższych macierzy, odejmij +4,000 * wiersz-1 z AM od wiersza-2 z AM i
20     odejmij +4.000 * wiersz-1 z IM od rzędu-2 z IM
21
22 AM Matrix IM Matrix
23 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
24 [' +0.000', ' -0.200', ' -0.400', ' -0.600', ' +4.200'] [' -0.800', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
25 [' +3.000', ' +2.000', ' +9.000', ' +5.000', ' +4.000'] [' +0.000', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
26 [' +2.000', ' +1.000', ' +5.000', ' +4.000', ' +3.000'] [' +0.000', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
27 [' +1.000', ' +2.000', ' +3.000', ' +4.000', ' +5.000'] [' +0.000', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
28
29 Korzystając z powyższych macierzy, odejmij +3,000 * wiersz-1 z AM od wiersza-3 z AM i
30     odejmij +3.000 * rząd-1 z IM od rzędu-3 z IM
31
32 AM Matrix IM Matrix
33 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
34 [' +0.000', ' -0.200', ' -0.400', ' -0.600', ' +4.200'] [' -0.800', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
35 [' +0.000', ' -0.400', ' +7.200', ' +3.800', ' +3.400'] [' -0.600', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
36 [' +2.000', ' +1.000', ' +5.000', ' +4.000', ' +3.000'] [' +0.000', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
37 [' +1.000', ' +2.000', ' +3.000', ' +4.000', ' +5.000'] [' +0.000', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
38
39 Korzystając z powyższych macierzy, odejmij +2,000 * wiersz-1 z AM od wiersza-4 z AM i
40     odejmij +2.000 * rząd-1 z IM od rzędu-4 z IM
41
42 AM Matrix IM Matrix
43 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
44 [' +0.000', ' -0.200', ' -0.400', ' -0.600', ' +4.200'] [' -0.800', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
45 [' +0.000', ' -0.400', ' +7.200', ' +3.800', ' +3.400'] [' -0.600', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
46 [' +0.000', ' -0.600', ' +3.800', ' +3.200', ' +2.600'] [' -0.400', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
47 [' +1.000', ' +2.000', ' +3.000', ' +4.000', ' +5.000'] [' +0.000', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
48
49 Korzystając z powyższych macierzy, odejmij +1,000 * wiersz-1 z AM od wiersza-5 z AM i
50     odejmij +1.000 * wiersz-1 z IM od wiersza-5 z IM
51
52 AM Matrix IM Matrix
53 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
54 [' +0.000', ' -0.200', ' -0.400', ' -0.600', ' +4.200'] [' -0.800', ' +1.000', ' +0.000', ' +0.000', ' +0.000 ']
55 [' +0.000', ' -0.400', ' +7.200', ' +3.800', ' +3.400'] [' -0.600', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
56 [' +0.000', ' -0.600', ' +3.800', ' +3.200', ' +2.600'] [' -0.400', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
57 [' +0.000', ' +1.200', ' +2.400', ' +3.600', ' +4.800'] [' -0.200', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
58
```

```

60 Korzystając z powyższych macierzy, przeskaluj rząd-2 AM i IM według elementu diagonalnego 2 AM, który wynosi 1/-0,200.
61
62 AM Matrix IM Matrix
63 [' +1.000', ' +0.800', ' +0.600', ' +0.400', ' +0.200'] [' +0.200', ' +0.000', ' +0.000', ' +0.000', ' +0.000 ']
64 [' -0.000', ' +1.000', ' +2.000', ' +3.000', ' -21.000'] [' +4.000', ' -5.000', ' -0.000', ' -0.000', ' -0.000 ']
65 [' +0.000', ' -0.400', ' +7.200', ' +3.800', ' +3.400'] [' -0.600', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
66 [' +0.000', ' -0.600', ' +3.800', ' +3.200', ' +2.600'] [' -0.400', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
67 [' +0.000', ' +1.200', ' +2.400', ' +3.600', ' +4.800'] [' -0.200', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
68
69 Korzystając z powyższych macierzy, odejmij +0,800 * wiersz-2 z AM od wiersza-1 z AM i
70     odjąć +0,800 * rząd-2 z IM od rzędu-1 z IM
71
72 AM Matrix IM Matrix
73 [' +1.000', ' +0.000', ' -1.000', ' -2.000', ' +17.000'] [' -3.000', ' +4.000', ' +0.000', ' +0.000', ' +0.000 ']
74 [' -0.000', ' +1.000', ' +2.000', ' +3.000', ' -21.000'] [' +4.000', ' -5.000', ' -0.000', ' -0.000', ' -0.000 ']
75 [' +0.000', ' -0.400', ' +7.200', ' +3.800', ' +3.400'] [' -0.600', ' +0.000', ' +1.000', ' +0.000', ' +0.000 ']
76 [' +0.000', ' -0.600', ' +3.800', ' +3.200', ' +2.600'] [' -0.400', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
77 [' +0.000', ' +1.200', ' +2.400', ' +3.600', ' +4.800'] [' -0.200', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
78
79 Korzystając z powyższych macierzy, odejmij -0,400 * rząd-2 z AM od wiersza-3 z AM i
80     odjąć -0,400 * rząd-2 z IM od rzędu-3 z IM
81
82 AM Matrix IM Matrix
83 [' +1.000', ' +0.000', ' -1.000', ' -2.000', ' +17.000'] [' -3.000', ' +4.000', ' +0.000', ' +0.000', ' +0.000 ']
84 [' -0.000', ' +1.000', ' +2.000', ' +3.000', ' -21.000'] [' +4.000', ' -5.000', ' -0.000', ' -0.000', ' -0.000 ']
85 [' +0.000', ' +0.000', ' +8.000', ' +5.000', ' -5.000'] [' +1.000', ' -2.000', ' +1.000', ' +0.000', ' +0.000 ']
86 [' +0.000', ' -0.600', ' +3.800', ' +3.200', ' +2.600'] [' -0.400', ' +0.000', ' +0.000', ' +1.000', ' +0.000 ']
87 [' +0.000', ' +1.200', ' +2.400', ' +3.600', ' +4.800'] [' -0.200', ' +0.000', ' +0.000', ' +0.000', ' +1.000 ']
88

```

Zadania do wykonania

Zadanie 1

Znajdź macierz odwrotną poniższej macierzy

**** Wykonaj sprawdzenie:** w tym celu zastosuj funkcję `np.linalg.inv()` oraz wykonaj mnożenie macierzy (podstawienie do wzoru) -> „natomiast zmodyfikowana macierz główna powinna mieć formę macierzy jednostkowej”

Zastosuj komentarze w celu opisanie kluczowych linijek kodu.

```

[8, 1, 4, 2, 1]
[8, 6, 4, 2, 1]
[1, 2, 3, 4, 1]
[8, 0, 6, 2, 5]
[1, 9, 6, 4, 1]

```