

BÁO CÁO ĐỒ ÁN 2

1 Các kí hiệu được sử dụng

- A (chữ hoa): ma trận cấp n hay ma trận với n dòng và n cột. Với phần tử $A[i][j]$ bất kì trong ma trận, ta có chỉ số dòng và chỉ số cột sẽ nằm trong khoảng sau: $i, j \in [0, n - 1]$
- $A[i]$: tất cả phần tử thuộc dòng i trong ma trận A
- \mathbf{x} (in đậm, chữ thường): vector x

2 Ý tưởng thực hiện

Ta rút ra một vài nhận xét trước khi bước vào ý tưởng thực hiện của hàm tìm nghịch đảo của ma trận như sau:

- A không khả nghịch \iff Tồn tại 1 dòng hay 1 cột của ma trận A bằng vector $\mathbf{0}$ (sử dụng định nghĩa về định thức để chứng minh)
- Khi ta sử dụng 3 phép biến đổi sơ cấp trên dòng/cột, định thức của ma trận không bị thay đổi
- Nếu tồn tại một dãy phép biến đổi sao cho $(A|I_n) \rightarrow \dots \rightarrow (I_n|B)$ thì $A = B^{-1}$ hay $B = A^{-1}$

Như vậy ta có thể thấy bản chất của việc tìm ma trận nghịch đảo chỉ là dùng phép khử Gauss–Jordan như đã thực hiện ở đồ án 1 để đưa ma trận ban đầu thành ma trận đơn vị. Nếu tồn tại 1 dãy phép biến đổi S làm ma trận ban đầu thành ma trận đơn vị thì ta áp dụng dãy S vừa làm vào I_n thì sẽ ra ma trận nghịch đảo. Ngược lại, nếu không thể đưa về ma trận đơn vị thì ta kết luận ma trận không khả nghịch.

Các bước thực hiện

- Khởi tạo ma trận B ban đầu là ma trận đơn vị I_n .
- Duyệt từng dòng từ trên xuống dưới trong ma trận A . Ta gọi chỉ số dòng mà thuật toán đang xét là i .
- Bước 1: Xét từ vị trí $A[i][i]$ trở xuống, ta thu được vector \mathbf{x} . Nếu $\mathbf{x} = \mathbf{0}$ thì ta kết luận không tồn tại ma trận nghịch đảo, ngược lại ta tiếp tục bước 2.
- Bước 2: Với cả 2 ma trận A (ma trận đầu vào) và B (ma trận vừa khởi tạo ở trên), đổi dòng i của cả 2 ma trận A và B nếu $A[i][i] = 0$ với dòng j có $A[j][i] \neq 0$ mà $j > i$.
- Bước 3: Chia $A[i]$ và $B[i]$ cho $A[i][i]$.
- Bước 4: $\forall j \in [0, n - 1], i \neq j, A[j] = A[j] - A[j][i] * A[i]$ và $B[j] = B[j] - A[j][i] * B[i]$

3 Mô tả hàm

```
1 def inverse(A):
2     n = len(A)
3     can_inv = True
4
5     res = [[1 if i == j else 0 for j in range(n)] for i in range(n)]
```

```

6
7     for i in range(n):
8         for j in range(i, n):
9             if A[j][i] != 0: break
10        if j == n - 1 and A[j][i] == 0:
11            can_inv = False
12            break
13        A[i], A[j] = A[j], A[i]
14        res[i], res[j] = res[j], res[i]
15        temp = A[i][i]
16        for k in range(n): A[i][k] /= temp
17        for k in range(n): res[i][k] /= temp
18        for j in range(n):
19            if j != i:
20                temp = A[j][i]
21                for k in range(n): A[j][k] -= temp * A[i][k]
22                for k in range(n): res[j][k] -= temp * res[i][k]
23
24    return can_inv, res

```

- Đầu tiên ta khởi tạo n là số dòng cũng như là số cột của A , khởi tạo biến `can_inv` để thông báo rằng A có khả nghịch hay không và khởi tạo ma trận `res` (ma trận nghịch đảo nếu có của A) ban đầu là ma trận đơn vị I_n .
- Tiếp theo ta duyệt theo từng dòng từ trên xuống dưới.
 - Kiểm tra từ dòng i đến dòng $n - 1$ trong cột i để xem có giá trị nào khác 0 hay không. Nếu có thì vector \mathbf{x} như mô tả ở trên khác vector $\mathbf{0}$.
 - Nếu kết thúc quá trình duyệt ở trên mà $j = n - 1 \rightarrow$ Từ dòng i đến dòng $n - 2$ của cột i ta có $A[i][i] = A[i+1][i] = \dots = A[n-2][i] = 0$. Ta xét giá trị của phần tử $A[j][i] = A[n-1][i]$.
 - * Nếu $A[j][i] = A[n-1][i] = 0 \Rightarrow \mathbf{x} = \mathbf{0} \Rightarrow A$ không khả nghịch \Rightarrow Kết thúc vòng lặp, `can_inv = False` (can't inverse this matrix) và kết thúc thuật toán.
 - * Ngược lại ta tiếp tục chạy thuật toán.
 - j là dòng đầu tiên làm $A[j][i] \neq 0$, đổi 2 dòng $A[i]$ và $A[j]$ với nhau cũng như đổi 2 dòng `res[i]` và `res[j]` (dòng 13 và 14) (nếu $i = j \rightarrow A$ và `res` không thay đổi gì).
 - 3 dòng tiếp theo (15 đến 17) thể hiện bước 3 của phần ý tưởng thuật toán. Ta chia dòng i của cả 2 ma trận A và `res` cho giá trị $A[i][i]$.
 - Dòng 18 đến dòng 22 thực hiện bước 4 như đã nêu ở trên.
- Hàm này sẽ output ra 2 giá trị: giá trị đầu tiên thể hiện cho việc ma trận có khả nghịch hay không (biến `can_inv`) và giá trị thứ 2 là ma trận nghịch đảo của ma trận đầu vào (nếu có) (biến `res`).