در روش تقسیم و حل برای معکوس گیری یک ماتریس مربعی $n\times n$ از ایده تقسیم ماتریس به ۴ ماتریس کوچکتر به‌صورت بلوکی استفاده می کنیم. فرض کنید ماتریس A به صورت زیر تقسیم شده باشد:

A=(A11 A21

A12 A22)

که در آن $A11, A12, A21, 22A$ هر کدام یک ماتریس $n/۲ * n/۲$ هستند. هدف یافتن معکوس A، یعنی یافتن ماتریسی B به گونه‌ای که: $A\times B=B\times A=I$ باشد.

ایده ی کلی استفاده از schur complement یا مکمل شر است.

برای این که بتونیم از تقسیم و حل استفاده کنیم، باید از رابطه‌ی مکمل شر استفاده کنیم. اگر $A۱۱$ معکوس‌پذیر باشد، میتونیم این کار را انجام بدیم:

۱.محاسبه معکوس زیرماتریس $A۱۱$:
$X = A۱۱^{-۱}$

۲.محاسبه‌ی چند ضرب ماتریسی:
ابتدا ماتریس Y را به صورت $Y=A۲۱\times X$ محاسبه می کنیم.

۳.محاسبه مکمل شر:
ماتریس مکمل شر را به این صورت تعریف میکنیم:

$S = A۲۲ - A۲۱\times X\times A۱۲$

و سپس معکوس آن را محاسبه می کنیم:

$Z=S^{-۱}$

۴.محاسبه‌ی ماتریس‌های میانی:
یک ماتریس واسطه W تعریف می کنیم:

$W=X\times A۱۲\times Z$

۵.نتایج را برای ساخت معکوس A ترکیب میکنیم:
در نهایت، معکوس A به صورت بلوکی به دست میاد :

$B = A^{-۱} = (X+W\times Y \quad -W$

$-Z\times Y \quad Z)$

```
Function InverseMatrix(A):
    if n == 1:
        return 1 / A[0][0]
    else:
        Partition A into A11, A12, A21, A22

        X = InverseMatrix(A11)
        Y = A21 * X
        S = A22 - A21 * X * A12
        Z = InverseMatrix(S)

        W = X * A12 * Z

        B11 = X + W * Y
        B12 = -W
        B21 = -Z * Y
        B22 = Z

        Combine B11, B12, B21, B22 into matrix B
        return B
```

تحلیل پیچیدگی زمانی

فرض میکنیم $T(n)$ زمان مورد نیاز برای محاسبه معکوس یک ماتریس $n \times n$ با استفاده از این الگوریتم باشه. در هر مرحله:

۱.دو فراخوانی بازگشتی برای معکوس‌گیری ماتریس‌های $n/2 * n/2$ داریم که میشه $2T(n/2)$.

۲.چند عمل ضرب و تفریق ماتریسی داریم که هر کدوم به‌طور معمول زمان $O(n^3)$ در ضرب ماتریسی استاندارد می‌برند (اگر از ضرب ماتریسی کلاسیک استفاده کنیم).

بنابراین رابطه بازگشتی به این صورت بدست میاریم:

$$T(n)=2T(n/2)+O(n^3)$$

با استفاده از تئوری ماستر، عبارت $O(n^3)$ قسمت غالب است، پس پیچیدگی زمانی کلی این الگوریتم برابر است با:

$$T(n)=O(n^3)$$

باید این را مد نظر داشته باشیم که اگر از الگوریتم استراسن استفاده کنیم، میتونیم پیچیدگی را به نزدیک $O(n^{2.8})$ برسونیم.

محمد کریمی ۴۰۲۳۶۱۳۰۶۰. امیررضا محمدی یگانه ۴۰۲۳۶۱۳۰۶۸.