AiSD – Zadanie 7 lista 3

Mateusz Szałowski (298976)

21 kwietnia 2020

Macierz A rozmiaru $n \times n$ nazywamy macierzą Toeplitza, jeśli jej elementy spełniają równanie $A[i, j] = A[i-1, j-1]$ dla $2 \leqslant i, j \leqslant n$.

To znaczy, że taka macierz ma takie same elementy na każdej przekątnej:

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & \dots & t_{1n} \\ t_{21} & t_{11} & t_{12} & \ddots & & \vdots \\ t_{31} & t_{21} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & t_{12} & t_{13} \\ \vdots & & \ddots & t_{21} & t_{11} & t_{12} \\ t_{n1} & \dots & \dots & t_{31} & t_{21} & t_{11} \end{bmatrix}$$

(a) Dodawanie macierzy Toeplitza w czasie $O(n)$

Zauważmy, że dodając 2 macierze $n \times n$ dodajemy każdą komórkę do każdej otrzymując nową macierz $n \times n$. W macierzy Toeplitza wiele tych komórek się powtarza, więc wystarczy, że policzymy wartość sumy tylko raz dla każdej unikatowej komórki.

Możemy reprezentować naszą macierz za pomocą jednowymiarowej tablicy. Na początku zapiszemy w niej cały pierwszy wiersz $t_{11}, t_{12}, t_{13}, \dots, t_{1n}$, a potem pierwszą kolumnę: $t_{21}, t_{31} \dots t_{n1}$ (pomijając element t_{11}, gdyż już się w tablicy znajduje).

Taka tablica ma długość $2n - 1$, a więc dodając dwie tablice do siebie wykonamy łącznie $2n - 1$ operacji dodawania, co daje nam złożoność $O(n)$

(b) Algorytm mnożenia macierzy Toeplitza oparty na metodzie "dziel i zwyciężaj"

Wejście: Macierz $T_{n\times n}$
Wyjście: Wektor $\vec{V}_{n\times 1}$

*Rozpatrzmy najpierw przypadek dla n parzystego, a następnie uogólnimy rozwiązanie.

Podzielmy naszą wejściową macierz Toeplitza na 4 ćwiartki oraz wektor V na 2 połówki:

$$\left[\begin{array}{ccc|ccc} t_{11} & t_{12} & t_{13} & \dots & \dots & t_{1n} \\ t_{21} & t_{11} & t_{12} & \ddots & & \vdots \\ t_{31} & t_{21} & \ddots & \ddots & \ddots & \vdots \\ \hline \vdots & \ddots & \ddots & \ddots & t_{12} & t_{13} \\ \vdots & & \ddots & t_{21} & t_{11} & t_{12} \\ t_{n1} & \dots & \dots & t_{31} & t_{21} & t_{11} \end{array}\right] \times \left[\begin{array}{c} \vec{v_1} \\ \vec{v_2} \\ \vec{v_3} \\ \hline \vdots \\ \vec{v_n} \end{array}\right]$$

$$\left[\begin{array}{c|c} A & B \\ \hline C & A \end{array}\right] \times \left[\begin{array}{c} \vec{V_1} \\ \hline \vec{V_2} \end{array}\right]$$

Wszystkie ćwiartki, które uzyskaliśmy również są macierzami Toeplitza. Zauważmy, że mamy 2 ćwiartki, które są identyczne (na rysunku oznaczone literą A).

oraz zachodzą zależności:

$$\begin{aligned} A\vec{V_1} &= A(\vec{V_1} + \vec{V_2} - \vec{V_2}) = A(\vec{V_1} + \vec{V_2}) - A(\vec{V_2}) \\ A\vec{V_2} &= A(\vec{V_2} + \vec{V_1} - \vec{V_1}) = A(\vec{V_1} + \vec{V_2}) - A(\vec{V_1}) \end{aligned}$$

Zatem nasz problem rekurencyjnego policzenia wyników dla 4 ćwiartek możemy zredukować następująco:

$$\left[\begin{array}{c|c} A & B \\ \hline C & A \end{array}\right] \times \left[\begin{array}{c} \vec{V_1} \\ \hline \vec{V_2} \end{array}\right] = \left[\begin{array}{c} A\vec{V_1} + B\vec{V_2} \\ A\vec{V_2} + C\vec{V_1} \end{array}\right] = \left[\begin{array}{c} A(\vec{V_1} + \vec{V_2}) - A\vec{V_2} + B\vec{V_2} \\ A(\vec{V_1} + \vec{V_2}) - A\vec{V_1} + C\vec{V_1} \end{array}\right] =$$

$$= \left[\begin{array}{c} A(\vec{V_1} + \vec{V_2}) - (A + B)\vec{V_2} \\ A(\vec{V_1} + \vec{V_2}) - (A + C)\vec{V_1} \end{array}\right]$$

Złożoność

W każdym wywołaniu rekurencyjnym wykonamy:

- 3 x rekurencyjne mnożenie macierzy $\{A(\vec{V_1}+\vec{V_2}),\ (A+B)\vec{V_2},\ (A+C)\vec{V_1}\} \to 3 \cdot T\left(\frac{n}{2}\right)$
- 2 x dodawanie macierzy Toeplitza $\frac{n}{2} \times \frac{n}{2}$ $\{A + B, A + C\} \to O(n)$
- 3 x dodawanie wektorów $\{\underbrace{\vec{V_1} + \vec{V_2}}_{1},\ \underbrace{A(\vec{V_1} + \vec{V_2}) - (A + B)\vec{V_2}}_{2},\ \underbrace{A(\vec{V_1} + \vec{V_2}) - (A + C)\vec{V_1}}_{3}\}$

  $\to 3 \cdot n = O(n)$

Zatem:

$$T(n) = 3 \cdot T\left(\frac{n}{2}\right) + O(n) =^1 O\left(n^{log_2(3)}\right) \approx O\left(n^{1.58}\right)$$

Uogólnienie dla n nieparzystego Jeśli w jakimś wywołaniu rekurencyjnym n jest nieparzyste, możemy rozszerzyć macierz $T_{n\times n} \to T'_{(n+1)\times(n+1)}$ przedłużając przekątne a w narożnikach dodając zera. Wektor V przedłużamy o jedno 0:

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & \dots & t_{1n} & 0 \\ t_{21} & t_{11} & t_{12} & \ddots & & \vdots & t_{1n} \\ t_{31} & t_{21} & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & t_{12} & t_{13} & \vdots \\ \vdots & & \ddots & t_{21} & t_{11} & t_{12} & t_{13} \\ t_{n1} & \dots & \dots & t_{31} & t_{21} & t_{11} & t_{12} \\ 0 & t_{n1} & \cdots & \cdots & t_{31} & t_{21} & t_{11} \end{bmatrix} \times \begin{bmatrix} \vec{v_1} \\ \vec{v_2} \\ \vec{v_3} \\ \vdots \\ \vec{v_n} \\ 0 \end{bmatrix}$$

Teraz gdy n już jest parzyste, wykonujemy nasz algorytm, a następnie z wektora wynikowego usuwamy ostatnią komórkę (która jest wynikiem mnożenia ostatniego wiersza rozszerzonej macierzy przez wektor).

[1] Z twierdzenia o rekurencji uniwersalnej

Skoro efekt sztucznie dodanego wiersza w wyniku jest już usunięty, pozostaje pytanie, czy sztucznie dodana kolumna nie zmieniła nam wyniku. Rozpiszmy więc jak wygląda k-ty wiersz wektora wynikowego:

$$t_{k1}\vec{v}_1 + t_{k2}\vec{v}_2 + \ldots + t_{kn}\vec{v}_n + t_{k(n+1)} \cdot 0$$

Ponieważ nasz wektor $\vec{V}$ rozszerzyliśmy o zero, każdy ostatni składnik sumy wektora wynikowego jest mnożony przez zero, a zatem wynik się nie zmienił.