Zadanie 5

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

- Podaj reprezentację macierzy Toeplitza, która umożliwia dodawanie dwóch takich macierzy w czasie $O(n)$.
- Podaj, oparty o metodę dziel i zwyciężaj, algorytm mnożenia macierzy Toeplitza przez wektor. Ile operacji arytmetycznych wymaga taki algorytm?

- Niech v_n będzie wektorem reprezentującym macierz Toeplitza (oznaczmy ją przez T) o rozmiarze $n \times n$, który ma postać $<T_{n1}, T_{n2}, \cdots, T_{n(n-1)}, T_{nn}, T_{(n-1)n}, \cdots, T_{1n}>$. Wektor ten ma rozmiar $2n-1$ i reprezentuje każdą przekątną (nie tylko główną) macierzy T. Dodawanie wektorów v_n wymaga $2n-1$ operacji arytmetycznych, co jest równe $O(n)$.

- Naszym zadaniem jest pomnożenie macierzy T przez wektor V. Macierz Toeplitza można przedstawić w postaci blokowej, tak samo jak wektor, przez który mnożymy. W takim razie rozważamy działanie

$$T \cdot V = T \cdot \begin{bmatrix} v_0 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} A & B \\ C & A \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

Zauważmy, że $AV_1 = A(V_1+V_2-V_2) = A(V_1+V_2)-AV_2$, oraz $AV_2 = A(V_1+V_2-V_1) = A(V_1+V_2)-AV_1$. W takim razie mamy

$$\begin{bmatrix} A & B \\ C & A \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} AV_1 + BV_2 \\ CV_1 + AV_2 \end{bmatrix} = \begin{bmatrix} A(V_1+V_2) - AV_2 + BV_2 \\ CV_1 + A(V_1+V_2) - AV_1 \end{bmatrix} = \begin{bmatrix} A(V_1+V_2) + (B-A)V_2 \\ A(V_1+V_2) + (C-A)V_1 \end{bmatrix}$$

Zauważmy, że macierze A, B, C również są macierzami Toeplitza. Każdy krok metody wymaga dodania do siebie dwóch macierzy Toeplitza o rozmiarze $\frac{n}{2} \times \frac{n}{2}$, dodanie dwóch wektorów o rozmiarze $\frac{n}{2}$ i wykonanie trzech rekurencyjnych mnożeń macierzy $\frac{n}{2} \times \frac{n}{2}$ przez wektor o rozmiarze $\frac{n}{2}$. Rozwiązując rekurencję $T(n) = 3 \cdot T(\frac{n}{2}) + O(n)$ otrzymujemy, że powyższy algorytm wymaga $O(n^{\log_2 3})$ działań arytmetycznych.

Ale co w przypadku, gdy n jest nieparzyste? Można bardzo łatwo to obejść - bierzemy macierz z wykreślonym wierszem $\lceil \frac{n}{2} \rceil$ i kolumną $\lceil \frac{n}{2} \rceil$, wykonujemy algorytm tak jak w przypadku n parzystego, a wynik tego kroku będzie postaci:

$$\begin{bmatrix} V_1 + V_1' \\ \prod_{i=1}^{n} T_{\lceil n/2 \rceil i} \cdot v_i \\ V_2 + V_2' \end{bmatrix}$$

$$V_1' = \begin{bmatrix} T_{1\lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \\ T_{2\lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \\ \vdots \\ T_{\lfloor n/2 \rfloor \lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \end{bmatrix} \qquad V_2' = \begin{bmatrix} T_{(\lceil n/2 \rceil + 1)\lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \\ T_{(\lceil n/2 \rceil + 2)\lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \\ \vdots \\ T_{n\lceil n/2 \rceil} \cdot v_{\lceil n/2 \rceil} \end{bmatrix}$$

Zostanie przeprowadzonych $O(n)$ dodatkowych operacji, więc w dalszym ciągu wykonujemy $O(n^{\log_2 3})$ działań arytmetycznych.