5 Dodatek. Baza przestrzeni generowanej przez dane wektory

5.1 Sformułowanie zadania, związek z rozwiązywaniem równań

Macierz A z m wierszami i n kolumnami, o wyrazach $a_{i,j}$ dla $i \leqslant m$ i $j \leqslant n$ nazywa się schodkową, jeżeli istnieje taki ciąg $j_1, \ldots, j_r$ liczb spełniających nierówności $1 \leqslant j_1 < \ldots < j_r \leqslant n$, że

1) $a_{i,j} = 0$ dla wszystkich $i = 1, \ldots, r$ i dla $j < j_i$,
2) $a_{i,j} = 0$ dla wszystkich $i > r$ i dla wszystkich $j = 1, \ldots, n$,
3) $a_{i,j_i} \neq 0$ dla wszystkich $i = 1, \ldots, r$.

Będziemy rozwiązywać następujące zadanie:

Zadanie 5.1 Sprowadzanie macierzy do postaci schodkowej.

Dane: macierz A o m wierszach i n kolumnach,

Rozmiar zadania: liczby m i n,

Znaleźć odwracalną macierz U o wymiarach $m \times m$ taką, że UA jest macierzą schodkową oraz iloczyn UA (często żąda się, aby dodatkowo $|\,U\,| = 1$).

Zauważmy, że wiersze macierzy UA są kombinacjami liniowymi wierszy macierzy A. Ponieważ dla odwracalnej macierzy U zachodzi także równość $A = U^{-1}(UA)$, więc w takim przypadku macierze A oraz UA generują tę samą przestrzeń. Jeżeli UA jest macierzą schodkową, to jej niezerowe wiersze są liniowo niezależne. Wobec tego, jeżeli rozwiążemy zadanie 5.1, to także znajdziemy szczególną bazę przestrzeni generowanej przez wiersze danej macierzy.

Wykorzystując zadanie 5.1 można rozwiązać zadanie 4.6 wykonując na przykład następujący algorytm:

Algorytm 5.1 znajdujący rozwiązanie układu równań liniowych

1) Przypuśćmy, że mamy dany układ $A\vec{x} = \vec{b}$ złożony z m równań z n zmiennymi. Najpierw uruchamiamy algorytm dla zadania 5.1. Przyjmijmy, że algorytm ten zwróci macierze U oraz $S = UA$,
2) Niech r będzie liczbą niezerowych wierszy macierzy S. Macierz S przedstawiamy w postaci $\begin{bmatrix} S_1 \\ 0 \end{bmatrix}$, gdzie S_1 jest macierzą o r (niezerowych, liniowo niezależnych) wierszach.
3) Obliczamy $\vec{c} = U\vec{b}$ i dzielimy go na dwie części: $\vec{c}_1$ długości r i resztę $\vec{c}_2$.
4) Sprawdzamy, czy $\vec{c}_2$ ma wszystkie współrzedne równe 0; jeżeli tak nie jest, to kończymy obliczenia stwierdzając, że dany układ równań jest sprzeczny i nie ma rozwiązania.
5) rozwiązujemy układ $S_1\vec{x} = \vec{c}_1$ uruchamiając np. algorytm dla zadania 4.5 i zwracamy rozwiązanie znalezione przez ten algorytm.

5.2 Zadania pomocnicze

Najpierw będziemy zajmować się kilkoma zadaniami pomocniczymi. Algorytmy rozwiązujące te zadania łatwiej sformułować przy założeniu, że dane macierze mają wymiary będące potęgami dwójki. Rozważane zadania charakteryzują się tym, że trudno je rozwiązać powiększając rozmiar danych (np. do potęgi dwójki), rozwiązując zadanie dla większych danych, a następnie przekazując część obliczonych wartości. Przedstawiając algorytmy będziemy przyjmować jednak dodatkowe założenia o rozmiarze danych. Aby rozwiązać zadanie w ogólnym przypadku wystarczy przekonać się, że dodatkowe założenia nie są istotne i w podobny sposób, komplikując opis, można algorytmy przekształcić w równie dobre, dzielące dane w przybliżeniu na pół.

Zadanie 5.2 Częściowe sprowadzanie do macierzy trójkątnych.

Dane: macierz A o $m \leqslant n$ wierszach i n kolumnach oraz górna trójkątna macierz R o wymiarach $n \times n$.

Rozmiar zadania: liczba n.

Znaleźć odwracalną macierz U oraz górną trójkątną macierz R' o wymiarach $n \times n$ taką, że

$$U \begin{bmatrix} A \\ R \end{bmatrix} = \begin{bmatrix} R' \\ 0 \end{bmatrix}.$$

Algorytm 5.2 rozwiązujący zadanie 5.2 (przyjmujemy dodatkowo, że n jest potęgą dwójki).

1) jeżeli $n = m = 1$, to $A = [a]$, $R = [r]$ dla pewnych a i r i wtedy przerywamy działanie algorytmu 5.2 zwracając macierz

$$U = \begin{bmatrix} 1 & 0 \\ -ra^{-1} & 1 \end{bmatrix} \text{ lub } U = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

zależnie od tego, czy $a \neq 0$, czy też $a = 0$, i odpowiednią macierz R';

2) jeżeli $m = 0$, to przerywamy działanie algorytmu 5.2 zwracając macierze $U = 1$ i $R' = R$.

3) Dalej będziemy zakładać, że jeszcze $m = n$. Jeżeli $m < n$, to postępujemy analogicznie posługując się mniejszymi macierzami. Macierz

$$\begin{bmatrix} A \\ R \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \\ R_1 & A_5 \\ 0 & R_2 \end{bmatrix}$$

dzielimy na 8 podmacierzy o wymiarach $n/2 \times n/2$ (zauważmy, że macierze R_1 i R_2 są trójkątne)

4) wywołujemy rekurencyjnie algorytm 5.2 dla macierzy A_3 i R_1, przypuśćmy, że wywołanie to zwraca macierze U_1 i R_3;

5) obliczamy
$$\begin{bmatrix} A_6 \\ A_7 \end{bmatrix} = U_1 \begin{bmatrix} A_4 \\ A_5 \end{bmatrix};$$

6) wywołujemy rekurencyjnie algorytm 5.2 dla macierzy A_1 i R_3, przypuśćmy, że wywołanie to zwraca macierz U_2 i R_4;

7) obliczamy
$$\begin{bmatrix} A_8 \\ A_9 \end{bmatrix} = U_2 \begin{bmatrix} A_2 \\ A_6 \end{bmatrix};$$

8) wywołujemy rekurencyjnie algorytm 5.2 dla macierzy A_7 i R_2, przypuśćmy, że wywołanie to zwraca macierze U_3 i R_5;

9) wywołujemy rekurencyjnie algorytm 5.2 dla macierzy A_9 i R_5, przypuśćmy, że wywołanie to zwraca macierz U_4 i R_6;

10) obliczamy i zwracamy (1 to macierz jednostkowa o wymiarach $n/2 \times n/2$)
$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} U_2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

11) zwracamy
$$R' = \begin{bmatrix} R_4 & A_8 \\ 0 & R_6 \end{bmatrix}.$$

Poprawność algorytmu 5.2 wynika z następujących rachunków:

$$U \begin{bmatrix} A \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} U_2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \\ R_1 & A_5 \\ 0 & R_2 \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} U_2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_1 & A_2 \\ R_3 & A_6 \\ 0 & A_7 \\ 0 & R_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} R_4 & A_8 \\ 0 & A_9 \\ 0 & A_7 \\ 0 & R_2 \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_4 & A_8 \\ 0 & A_9 \\ 0 & R_5 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_4 & A_8 \\ 0 & R_6 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R' \\ 0 \end{bmatrix}.$$

Niech α będzie większe od 2 i niech $M(n) = c_1 \cdot n^\alpha$ będzie złożonością algorytmu mnożenia macierzy o wymiarach $n \times n$. Jeżeli $f(n)$ oznacza liczbę operacji arytmetycznych wykonywanych przez wyżej algorytm 5.2, to $f(1) = 1$ oraz

$$f(n) \leqslant 4 \cdot f(\frac{n}{2}) + c_2 \cdot M(\frac{n}{2}) = 4 \cdot f(\frac{n}{2}) + c \cdot n^\alpha$$

dla pewnej stałej c. Z tej nierówności otrzymujemy, że

$$f(n) \leqslant \frac{2^\alpha \cdot c}{2^\alpha - 4} \cdot (n^\alpha - n^2) + n^2$$

dla wszystkich n będących potęgą dwójki i – w konsekwencji – $f(n) \leqslant c_3 \cdot M(n)$. Pierwszą z dwóch ostatnich nierówności dowodzimy przez indukcję w następujący sposób:

$$f(n) \leqslant 4 \cdot \left[\frac{2^\alpha \cdot c}{2^\alpha - 4} \cdot ((\frac{n}{2})^\alpha - (\frac{n}{2})^2) + (\frac{n}{2})^2\right] + c \cdot n^\alpha =$$
$$\frac{4 \cdot c}{2^\alpha - 4} \cdot n^\alpha - \frac{2^\alpha \cdot c}{2^\alpha - 4} \cdot n^2 + \frac{(2^\alpha - 4) \cdot c}{2^\alpha - 4} \cdot n^\alpha + n^2 = \frac{2^\alpha \cdot c}{2^\alpha - 4} \cdot (n^\alpha - n^2) + n^2.$$

Zadanie 5.3 Sprowadzanie do macierzy trójkątnych.

Dane: macierz kwadratowa A o n wierszach.

Rozmiar zadania: liczba n.

Znaleźć odwracalną macierz U oraz górną trójkątną macierz R o wymiarach $n \times n$ taką, że $UA = R$.

Algorytm 5.3 rozwiązujący zadanie 5.3 (dodatkowo zakładamy, że n jest potęgą dwójki).

1) jeżeli $n = 1$, to $A = [a]$ dla pewnego a i wtedy przerywamy działanie algorytmu 5.3 zwracając macierze $U = [1]$ i A.

2) macierz A dzielimy na cztery podmacierze

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix};$$

3) wywołujemy rekurencyjnie algorytm 5.3 dla macierzy A_3, przypuśćmy, że wywołanie to zwraca macierze U_1 i R_1;

4) obliczamy $A_5 = U_1 A_4$;

5) wywołujemy algorytm 5.2 dla macierzy A_1 i R_1, przypuśćmy, że wywołanie to zwraca macierz U_2 i R_2;

6) obliczamy

$$\begin{bmatrix} A_6 \\ A_7 \end{bmatrix} = U_2 \begin{bmatrix} A_2 \\ A_5 \end{bmatrix};$$

7) wywołujemy rekurencyjnie algorytm 5.3 dla macierzy A_7, przypuśćmy, że wywołanie to zwraca macierze U_3 i R_3;

8) obliczamy i zwracamy (1 to macierz jednostkowa o wymiarach $n/2 \times n/2$)

$$U = \begin{bmatrix} 1 & 0 \\ 0 & U_3 \end{bmatrix} U_2 \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}.$$

9) zwracamy

$$R' = \begin{bmatrix} R_2 & A_6 \\ 0 & R_3 \end{bmatrix}.$$

Poprawność algorytmu 5.3 wynika z następujących rachunków:

$$UA = \begin{bmatrix} 1 & 0 \\ 0 & U_3 \end{bmatrix} U_2 \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_3 \end{bmatrix} U_2 \begin{bmatrix} A_1 & A_2 \\ R_1 & A_5 \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & U_3 \end{bmatrix} \begin{bmatrix} R_2 & A_6 \\ 0 & A_7 \end{bmatrix} = \begin{bmatrix} R_2 & A_6 \\ 0 & R_3 \end{bmatrix} = R.$$

Jeżeli teraz $f_2(n)$ oznacza liczbę operacji arytmetycznych wykonywanych przez algorytm 5.3, to $f_2(1) = 0$ oraz

$$f_2(n) \leqslant 2 \cdot f_2(\frac{n}{2}) + f_1(\frac{n}{2}) + c_2 \cdot M(\frac{n}{2}) \leqslant 2 \cdot f_2(\frac{n}{2}) + c \cdot n^\alpha$$

dla pewnej stałej c. Z tej nierówności otrzymujemy, że

$$f_2(n) \leqslant \frac{2^\alpha \cdot c}{2^\alpha - 2} \cdot (n^\alpha - n)$$

dla wszystkich n będących potęgą dwójki i – w konsekwencji – $f_2(n) \leqslant c_4 \cdot M(n)$. Pierwszą z tych nierówności dowodzimy przez indukcję tak, jak w poprzednim przypadku:

$$f_2(n) \leqslant 2 \cdot \frac{2^\alpha \cdot c}{2^\alpha - 2} \cdot ((\frac{n}{2})^\alpha - \frac{n}{2}) + c \cdot n^\alpha =$$
$$\frac{2 \cdot c}{2^\alpha - 2} \cdot n^\alpha - \frac{2^\alpha \cdot c}{2^\alpha - 2} \cdot n + \frac{(2^\alpha - 2) \cdot c}{2^\alpha - 2} \cdot n^\alpha = \frac{2^\alpha \cdot c}{2^\alpha - 2} \cdot (n^\alpha - n).$$

Zadanie 5.4 Sprowadzanie macierzy trójkątnej do macierzy schodkowej.

Dane: kwadratowa trójkątna górna macierz R o n wierszach.

Rozmiar zadania: liczba n.

Znaleźć odwracalną macierz U oraz schodkowa macierz S o wymiarach $n \times n$ taką, że $UR = S$.

Algorytm 5.4 rozwiązujący zadanie 5.4 (zakładamy, że n jest potęgą dwójki).

1) jeżeli $n = 1$, to $R = [a]$ dla pewnego a i wtedy przerywamy działanie algorytmu 5.4 zwracając macierze $U = [1]$ i R.

2) macierz R dzielimy na cztery podmacierze

$$R = \begin{bmatrix} R_1 & A_1 \\ 0 & R_2 \end{bmatrix};$$

3) wywołujemy rekurencyjnie algorytm 5.4 dla macierzy R_1, przypuśćmy, że wywołanie to zwraca macierze U_1 i macierz schodkową złożoną z macierzy S_1 o niezerowych wierszach i kilku wierszy zerowych;

4) obliczamy $\begin{bmatrix} A_2 \\ A_3 \end{bmatrix} = U_1 A_1$, przy czym A_2 ma tyle samo wierszy, co S_1;

5) wywołujemy algorytm 5.2 dla macierzy A_3 i R_2, przypuśćmy, że wywołanie to zwraca macierz U_2 i R_3;

6) wywołujemy rekurencyjnie algorytm 5.4 dla macierzy R_3, przypuśćmy, że wywołanie to zwraca macierze U_3 i S_2;

7) obliczamy i zwracamy

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & 1 \end{bmatrix}.$$

8) zwracamy

$$S = \begin{bmatrix} S_1 & A_2 \\ 0 & S_2 \\ 0 & 0 \end{bmatrix}.$$

Poprawność algorytmu 5.4 wynika z następujących rachunków:

$$UR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & A_1 \\ 0 & R_2 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & U_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & A_2 \\ 0 & A_3 \\ 0 & R_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_1 & A_2 \\ 0 & R_3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} S_1 & A_2 \\ 0 & S_2 \\ 0 & 0 \end{bmatrix} = S.$$

Jeżeli teraz $f_3(n)$ oznacza liczbę operacji arytmetycznych wykonywanych przez algorytm 5.4, to $f_3(1) = 0$ oraz

$$f_3(n) \leqslant 2 \cdot f_3(\frac{n}{2}) + f_1(\frac{n}{2}) + c_5 \cdot M(\frac{n}{2}) \leqslant 2 \cdot f_3(\frac{n}{2}) + c \cdot n^\alpha$$

dla pewnej stałej c. Z tej nierówności, podobnie jak w poprzednim przypadku, otrzymujemy, że
$$f_2(n) \leqslant \frac{2^\alpha \cdot c}{2^\alpha - 2} \cdot (n^\alpha - n)$$
dla wszystkich n będących potęgą dwójki i – w konsekwencji – $f_2(n) \leqslant c_6 \cdot M(n)$.

5.3 Sprowadzanie macierzy do postaci schodkowej

Zadanie 5.5 Sprowadzanie do postaci schodkowej macierzy o krótkich wierszach.

Dane: macierz A o m wierszach i $n \leqslant m$ kolumnach,

Rozmiar zadania: liczba m,

Znaleźć odwracalną macierz U o wymiarach $m \times m$ oraz macierz schodkową S taką, że $UA = S$.

Algorytm 5.5 rozwiązujący zadanie 5.5.

1) Tworzymy kwadratową macierz $A_1 = [A, 0]$;

2) uruchamiamy dla macierzy A_1 algorytm 5.3; przypuśćmy, że algorytm ten zwrócił macierze U_1 i R;

3) uruchamiamy dla macierzy R algorytm 5.4; przypuśćmy, że algorytm ten zwrócił macierze U_2 i S;

4) Znajdujemy macierz S_1 składającą się z n pierwszych kolumn macierzy S i zwracamy macierze $U = U_1U_2$ oraz S_1.

Zadanie 5.6 Sprowadzanie dowolnej macierzy do postaci schodkowej.

Dane: macierz A o m wierszach i n kolumnach,

Rozmiar zadania: liczby m i n,

Znaleźć odwracalną macierz U o wymiarach $m \times m$ oraz macierz schodkową S taką, że $UA = S$.

Algorytm 5.6 rozwiązujący zadanie 5.6.

1) $Y := A$, $U := 1$, $k := m$, X i S są macierzami pustymi (zmienna k pamięta liczba wierszy macierzy A, która ma być przekształcana, cała przekształcana macierz jest pamiętana jako $[S \;\; X \;\; Y]$, macierz S zawiera część już przekształconą i jest schodkowa, macierze X i Y zawierają przekształconą w różnym stopniu resztę),

2) dopóki k ostatnich wierszy macierzy $[X \; Y]$ zawiera niezerowy element

   (a) jeżeli X ma mniej niż k kolumn, to (a.1) $l := \min\{m, \text{liczba kolumn macierzy } Y\}$,

(a.2) dzielimy macierz Y na dwie części: macierz Y_1 o l kolumnach i resztę Y_2 (tak więc $Y = [Y_1\ Y_2]$);

(a.3) $X := [X\ \ UY_1]$ oraz $Y := Y_2$

(b) $p := \min\{k, \text{liczba kolumn macierzy } X\}$,

(c) dzielimy macierz X na dwie części: macierz X_1 o p kolumnach i resztę X_2 (tak więc $X = [X_1\ X_2]$);

(d) $X := X_2$,

(e) przedstawiamy macierz X_1 w postaci $\begin{bmatrix} X_3 \\ X_4 \end{bmatrix}$, gdzie X_4 ma k wierszy,

(f) jeżeli macierz X_4 jest zerowa, to $S := [S\ \ X_1]$ (i zaczynamy wykonywać pętlę dopóki kolejny raz od początku), a w przeciwnym razie

(g) uruchamiamy dla macierzy X_4 algorytm 5.5; przypuśćmy, że algorytm ten zwrócił U_2 i macierz schodkową $S_1 = \begin{bmatrix} S_2 \\ 0 \end{bmatrix}$, gdzie S_2 jest macierzą o niezerowych wierszach;

(h) $X := \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} X$ oraz $U := \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} U$,

(i) $k := k -$ liczba wierszy macierzy S_2,

(j) $S := \left[S\ \ \begin{bmatrix} X_3 \\ S_2 \\ 0 \end{bmatrix} \right]$

3) zwracamy macierze U oraz $[S\ \ X\ \ UY]$.

Nietrudno zauważyć, że suma ilości kolumn macierzy S, X i Y jest niezmiennikiem pętli dopóki w algorytmie 5.6 i jest równa liczbie kolumn danej macierzy A. Jeżeli zmienna k ma wartość 0, to warunek sterujący pętlą staje się fałszywy i algorytm 5.6 zatrzymuje się. Jeżeli zaś na początku pętli $k > 0$, to także zmienna p przyjmuje wartość dodatnią. Ponieważ podczas każdego wykonania pętli zwiększa się o p liczba kolumn macierzy S, więc algorytm 5.6 kończy pracę dla dowolnych danych.

Przypuśćmy, że $k_1, k_2, \ldots, k_t$ są wartościami zmiennej k bezpośrednio po rozpoczęciu wykonywania kolejnych pętli, a $p_1, p_2, \ldots, p_t$ – odpowiadającymi im wartościami zmiennej p. Oczywiście, $p_i \leqslant k_i \leqslant m$ dla $i \leqslant t$. Jeżeli tuż po zdefiniowaniu wartości p będzie prawdziwa nierówność $p_i < k_i$, to liczba kolumn macierzy $[X_2\ \ Y]$ jest równa 0, czyli macierz $[X_2\ \ Y]$ jest pusta. W tej sytuacji algorytm 5.6 wykonuje pętlę po raz ostatni. Wobec tego, $p_i = k_i$ dla wszystkich $i < t$. Z poprzedniego akapitu wynika, że po zakończeniu działania algorytmu 5.6 macierz S ma $\sum_{i=1}^{t} p_i$ kolumn. W ten sposób otrzymujemy, że $\sum_{i=1}^{t} p_i \leqslant n$. Wobec tego prawdziwa jest także nierówność $\sum_{i=1}^{t} k_i \leqslant n + m$.

Łatwo zauważyć, że stwierdzenie „k ostatnich wierszy macierzy S składa się z samych zer" jest niezmiennikiem pętli w algorytmie 5.6 i – w konsekwencji – stwierdzenie to jest prawdziwe na początku i na końcu każdej pętli.

Przyjmijmy, że wyrażenie $c \cdot m^\alpha$ szacuje z góry liczbę operacji arytmetycznych wykonywanych zarówno podczas mnożenia macierzy kwadratowych o m wierszach, jak i podczas realizacji algorytmu 5.5 dla macierzy o takiej liczbie wierszy. Zauważmy także, że instrukcja wykonywana w pętli dopóki ma postać

$$\text{if war}_1 \text{ then instr}_1\text{; instr}_2\text{; if war}_2 \text{ then instr}_3 \text{ else instr}_4.$$

Podczas wykonania algorytmu 5.6 operacje arytmetyczne są wykonywane tylko podczas wykonywania instrukcji instr_1 lub instr_4, oraz po wykonaniu pętli dopóki. Zauważmy, że obliczając iloczyn

$$\begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} U,$$

gdzie macierz U_2 ma wymiary $k \times k$, a macierz $U - m \times m$, możemy ostatnie k wierszy macierzy U podzielić na macierze o wymiarach $k \times k$ i każdą otrzymaną w ten sposób macierz pomnożyć przez U_2. Wtedy cały iloczyn składa się z $m - k$ pierwszych wierszy macierzy U i obliczonych iloczynów o wymiarach $k \times k$. Podobnie możemy obliczać drugi iloczyn z punktu (h). Wobec tego, podczas wykonywania pojedyńczej instrukcji instr_4 jest realizowane

$$c \cdot k^\alpha + c \cdot \left\lceil \frac{m}{k} \right\rceil k^\alpha + c \cdot \left\lceil \frac{m}{k} \right\rceil k^\alpha \leqslant (3 \cdot c + 2) \cdot m^{\alpha-1} \cdot k$$

operacji arytmetycznych (pamiętajmy, że $k \leqslant m$). Podczas całego algorytmu 5.6, wykonanie instrukcji instr_4 wymaga

$$\sum_{i=1}^{t} (3 \cdot c + 2) \cdot m^{\alpha-1} \cdot k_i \leqslant (3 \cdot c + 2) \cdot m^{\alpha-1} \cdot \sum_{i=1}^{t} k_i \leqslant (3 \cdot c + 2) \cdot m^{\alpha-1} \cdot (n + m) \leqslant$$
$$\leqslant (6 \cdot c + 4) \cdot m^{\alpha-1} \cdot \max(n, m)$$

operacji arytmetycznych.

Zrealizowanie wszystkich instrukcji instr_1 i obliczenia kończące wymagają łącznie wykonania $\left\lceil \frac{n}{m} \right\rceil$ mnożeń macierzy kwadratowych o wymiarach $m \times m$. Uwzględniając fakt, że za pierwszym razem mnożymy przez macierz $U = 1$, te fragmenty algorytmu wymagają wykonania $c \cdot m^{\alpha-1} \cdot n$ operacji arytmetycznych. Przeprowadzone obliczenia dowodzą, że algorytm 5.6 uruchomiony z macierzą o m wierszach i n kolumnach wykonuje najwyżej

$$(7 \cdot c + 4) \cdot m^{\alpha-1} \cdot \max(n, m)$$

operacji arytmetycznych.

Pokażemy jeszcze, że równość $UA = [S \quad X \quad UY]$ jest niezmiennikiem pętli algorytmu 5.6. Oczywiście, równość ta jest prawdziwa przed pierwszym wykonaniem pętli. Jeżeli jest niezmiennikiem, jest też prawdziwa po wykonaniu pętli. Poprawność algorytmu 5.6 jest konsekwencją tego faktu i spostrzeżenia, że macierz $[S \;\; X \;\; UY]$ jest schodkowa. Tak jest, gdyż S jest schodkowa i k jej ostatnich wierszy składa się z samych zer, a z warunku zakończenia pętli wynika, że tę samą własność ma k ostatnich wierszy macierzy $[X \;\; Y]$, oraz – w konsekwencji – macierzy $[X \;\; UY]$.

Aby dowieść, że podana równość jest niezmiennikiem przyjmijmy, że indeks p oznacza początkową wartość zmiennej, a indeks k – końcową. Mamy więc

$$U_pA = [S_p \;\; X_p \;\; U_pY_p] = [S_p \;\; X_p \;\; U_p[Y_1 \;\; Y_2]] = [S_p \;\; X_p \;\; U_pY_1 \;\; U_pY_2] =$$
$$= [S_p \;\; [X_p \;\; U_pY_1] \;\; U_pY_2] = [S_p \;\; [X_1 \;\; X_2] \;\; U_pY_k] = [S_p \;\; \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} \;\; X_2 \;\; U_pY_k].$$

Jeżeli powyższą równość pomnożymy przez macierz $U_3 = \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix}$ i skorzystamy z definicji U_k, to otrzymamy

$$U_kA = [U_3S_p \;\; \begin{bmatrix} X_3 \\ U_2X_4 \end{bmatrix} \;\; U_3X_2 \;\; U_3U_pY_k] = [S_p \;\; \begin{bmatrix} X_3 \\ S_2 \\ 0 \end{bmatrix} \;\; X_k \;\; U_kY_k] = [S_k \;\; X_k \;\; U_kY_k].$$

Z powyższych rozważań wynika

Twierdzenie 5.1 (Keller-Gehrig) *Zadanie 5.6 można rozwiązać dla danej macierzy A o m wierszach i n kolumnach wykonując $c \cdot m^{\alpha-1} \cdot \max(n, m)$ operacji arytmetycznych, gdzie c jest pewną stałą, a α jest wykładnikiem z oszacowania złożoności algorytmu mnożenia macierzy.* □

Wniosek 5.2 *Dowolny układ m równań liniowych z n niewiadomymi można rozwiązać (można rozwiązać zadanie 4.6) wykonując $c \cdot m^{\alpha-1} \cdot \max(n, m)$ operacji arytmetycznych, gdzie c jest pewną stałą, a α jest wykładnikiem z oszacowania złożoności algorytmu mnożenia macierzy.*

Dowód. Wniosek wynika z analizy algorytmu 5.1, który w odpowiednich momentach uruchamia algorytm 5.6 i algorytm o odpowiedniej złożoności rozwiązujący zadanie 4.5 (np. algorytm przedstawiony w rozdz. 4). □