1 Wielomiany

1.1 Postac potegowa (naturalna) wielomianu

Wielomian moze byc zapisany w bazie: $\{1, x, x^2, \ldots, x^n\}$.

$$w(x) = \sum_{k=0}^{n} a_k x^k \tag{1}$$

Algorytm:

$$\begin{aligned} &s = a_0; \\ &xp = 1; \\ &for \quad i = 1 \quad to \quad n \quad begin \\ &xp = xp * x; \\ &s = s + a_i * xp \\ &end \end{aligned} \tag{2}$$

Liczba dzialan $\approx 2n(*) + n(+)$.

1.2 Algorytm Hornera

Aby policzyc wartosc wielomianu mozna zastosowac szybszy algorytm. Korzystamy z postaci:

$$w(x) = (...(a_n x + a_{n-1})x + ... + a_1)x + a_0 \tag{3}$$

Algorytm:

$$\begin{aligned} &w_n = a_n \\ &w_i(x) = w_{i+1}(x)x + a_i \quad dla \quad i = n-1, n-2, \ldots, 0 \\ &w(x) = w_0(x) \end{aligned} \tag{4}$$

Liczba dzialan $\approx n(*) + n(+)$.

Twierdzenie:
Algorytm Hornera jest jedynym algorytmem, ktory minimalizuje liczbe dodawan i mnozen przy obliczaniu wartosci wielomianu danego w postaci potegowej.

1.3 Dzielenie wielomianu $w(x)$ przez $(x - \alpha)$

Algorytm Hornera moze byc wykorzystany w celu podzielenia wielomianu $w(x)$ przez $(x - \alpha)$. Jezeli:

$$\sum_{k=0}^{n} a_k x^k = (\sum_{k=0}^{n-1} b_{k+1} x^k)(x - \alpha) + b_0, \tag{5}$$

gdzie
$(\sum_{k=0}^{n-1} b_{k+1} x^k)$ - wynik dzielenia,
b_0 - reszta z dzielenia.

To:

$$\begin{aligned} &a_n = b_n \\ &a_k = b_k - b_{k+1}\alpha \quad dla \quad k = n-1, n-2, \ldots, 0 \end{aligned} \tag{6}$$

Porownujac z algorytmem Hornera $(a_i = w_i(x) - w_{i+1}(x))$ $b_i = w_i(\alpha)$.

1.4 Postac Newtona wielomianu

Wielomian moze byc rowniez zapisany w bazie: $\{1, x-x_0, (x-x_0)(x-x_1), \ldots, (x-x_0)\cdots(x-x_n)\}$, gdzie $x_0, x_1, \ldots, x_n$ sa danymi liczbami.

$$w(x) = b_0 + \sum_{i=1}^{n} b_i(x-x_0)\cdots(x-x_i) \tag{7}$$

Algorytm:

$$\begin{aligned}
&s = b_0;\\
&xp = 1;\\
&for \quad i=1 \quad to \quad n \quad begin\\
&xp = xp*(x-x_{i-1});\\
&s = s + b_i * xp\\
&end
\end{aligned} \tag{8}$$

Liczba dzialan $\approx 2n(*) + 2n(+)$.

1.5 Uogolniony algorytm Hornera

Ponownie mozna zastosowac szybsza metode. Korzystamy z postaci:

$$w(x) = (...(b_n(x-x_{n-1}) + b_{n-1})(x-x_{n-2}) + ... + b_1)(x-x_0) + b_0 \tag{9}$$

Algorytm:

$$\begin{aligned}
&w_n = b_n\\
&w_i(x) = w_{i+1}(x-x_i)x + b_i \quad dla \quad i = n-1, n-2, \ldots, 0\\
&w(x) = w_0(x)
\end{aligned} \tag{10}$$

Liczba dzialan $\approx n(*) + 2n(+)$.

1.6 Przejscie od postaci Newtona do postaci potegowej

Znajac wspolczynniki wielomianu w postaci Newtona mozna wyznaczyc wspolczynniki jego postaci naturalnej. W tym celu kazdy z wielomianow pomocniczych zdefiniowanych wzorami 11 zapiszmy w postaci naturalnej:

$$w_i(x) = \sum_{k=i}^{n} a_k^{(i)} x^{k-i}, \quad i = n, n-1, \ldots 0 \tag{11}$$

Ze wzoru 11 otrzymujemy zaleznosci:

$$a_n^{(n)} = b_n \tag{12}$$

oraz

$$\begin{aligned}
w_i(x) \quad &= \sum_{k=i+1}^{n} a_k^{(i+1)} x^{k-(i+1)}(x-x_i) + b_i = \sum_{k=i+1}^{n} a_k^{(i+1)} x^{k-i} - \sum_{k=i+1}^{n} a_k^{(i+1)} x^{k-(i+1)} x_i + b_i = \\
&a_n^{(i+1)} x^{n-i} + \sum_{k=i+1}^{n-1} (a_k^{(i+1)} - a_{k+1}^{(i+1)} x_i) x^{k-i} + (b_i - a_{i+1}^{(i+1)} x_i).
\end{aligned} \tag{13}$$

Stad wspolczynniki postaci potegowej kolejnych wielomianow w_i $(i = n, n-1, \ldots, 0)$ mozna wyznaczyc rekurencyjnie:

$$\begin{aligned}
a_n^{(i)} &= a_n^{(i+1)} = \ldots = a_n^{(n)} = b_n\\
a_i^{(i)} &= b_i - a_{i+1}^{(i+1)} x_i\\
a_k^{(i)} &= a_k^{(i+1)} - a_{k+1}^{(i+1)} x_i, \quad dla \quad k = i+1, i+2, \ldots, n-1
\end{aligned} \tag{14}$$

Pamietajac, ze $w(x) = w_0(x)$ wiemy, ze szukane wspolczynniki a_k sa rowne $a_k^{(0)}$ $(k = 0, 1, \ldots, n)$. Daje to nastepujacy algorytm na przejscie od postaci Newtona do postaci potegowej:

$$
\begin{array}{l}
a[n] = b[n]; \\
for \quad i = n-1 \quad downto \quad 0 \quad do \\
begin \\
\quad xi = x[i]; \\
\quad a[i] = b[i]; \\
\quad for \quad k = i \quad to \quad n-1 \quad do \\
\quad a[k] = a[k] - xi * a[k+1]; \\
end
\end{array}
\tag{15}
$$

1.7 Tablice

Deklaracja:
int a[10];
definiuje zawsze tablice a[0..9]. Nie istnieje w C metoda zapisania wprost wektora o dowolnie zaczynajacym sie wskazniku (np. a[5..14]). W wielu przypadkach przydaja sie jednak tablice a[1..10]. Mozna je latwo zapisac przy pomocy wskaznikow:
int a[10],*ap;
ap=a-1;
Wskaznik *ap* wskazuje na pozycje przed *a* dzieki czemu stworzylismy tablice ap[1..10]. Mozna nastepnie wykonywac na niej dzialania.