

Dominika Maciąg

Dokumentacja: temat projektu nr 2.

Opis problemu i opis rozwiązania (w pseudokodzie)

Problem: w jaki sposób zaimplementować funkcję dodającą dowolne dwa wielomiany, które reprezentujemy za pomocą tablicy A[] oraz B[].

```
Funkcja(A[], B[], n, m) { //n, m – wielkości tablic wielomianów
    int maxl = max(n,m); //wywołujemy funkcję która da nam największą wartość
    int suma[maxl]; //tworzymy pomocniczą tablicę
    for(i=0; i<n; i++){
        suma[i]=A[i]; //wypełniamy wartościami wielomianu A
    }

    for(i=0; i<m; i++){
        //do naszej tablicy wypełnionej wartościami A dodajemy wartości B
        suma[i]+=B[i]
    }
}
```

Problem: w jaki sposób zaimplementować funkcję odejmującą dowolne dwa wielomiany.

Problem ten rozwiązujemy analogicznie do przypadku dodawania.

```
Funkcja(A[], B[], n, m) { //n, m – wielkości tablic wielomianów
    int maxl = max(n,m); //wywołujemy funkcję która da nam największą wartość
    int roznica[maxl]; //tworzymy pomocniczą tablicę
    for(i=0; i<n; i++){
        roznica[i]=A[i]; //wypełniamy wartościami wielomianu A
    }

    for(i=0; i<m; i++){
        //do naszej tablicy wypełnionej wartościami A odejmujemy wartości B
        roznica[i]-=B[i]
    }
}
```

Problem: w jaki sposób zaimplementować funkcję mnożącą dowolne dwa wielomiany.

```
Funkcja(A[], B[], n, m){ //n, m – wielkości tablic wielomianów
    int iloczyn[n+m-1]; //tworzymy pomocniczą tablicę

    //mnożymy każdy element A z każdym elementem B i je sumujemy
    for(i=0; i<n; i++){
        for(j=0; j<m; j++){
            iloczyn[j+i]+=B[j]*A[i];
        }
    }
}
```

```
}  
}
```

Problem: w jaki sposób obliczyć wartość wielomianu $W[]$ dla podanego x – aby to zrobić musimy zaimplementować funkcję która wykorzystuje algorytm Hornera

//argumenty funkcji po kolei to: wielomian, stopień wielomianu, podany x

```
Funkcja(W[], int n, int x){
```

```
    int wynik[]=W[n]; //tworzymy pomocniczą tablicę, przypisujemy jej element W[n]
```

```
    for(i=n-1; i>=0; i--){
```

```
        wynik=wynik*x+wielomian[i];
```

```
    }
```

```
}
```

Opis użytych struktur danych

Użyte w projekcie struktury danych to tablice.

Tablica to struktura danych, która przechowuje zbiór elementów tego samego typu.

Elementy te mają indeksy, dzięki czemu łatwo je przetwarzać w pętlach. Indeksy w tablicach zaczynają się od 0 do podanej przez nas wcześniej maksymalnej liczby elementów.

Oszacowanie złożoności czasowej i pamięciowej użytych struktur danych i podstawowych operacji na tych strukturach danych:

Dla tablicy o wymiarze n pamięć będzie wymagała $O(n)$ miejsca

Złożoności czasowe:

Deklaracja tablicy – $O(1)$

Dodawanie do tablicy – $O(n)$

Pętla – $O(n)$

Zwrócenie wartości – $O(1)$

Dwie zagnieżdżone pętle – $O(n^2)$

Oszacowanie złożoności czasowej i pamięciowej głównych algorytmów

Dla mojej implementacji algorytm Hornera ma złożoność czasową wynoszącą $O(n^2)$

Dokumentacja użytkowa: w jaki sposób uruchomić program oraz jak wprowadzić dane

Aby uruchomić program należy wpisać w konsolę:

```
g++ projekt.cpp -o projekt
```

```
./projekt
```

Dane wprowadzamy na początku naszego programu, tam wypełniamy zadeklarowaną tablicę $A[]$ oraz $B[]$, które reprezentują wielomiany.

Indeks 0 reprezentuje wyraz wolny wielomianu, indeks 1 reprezentuje wyraz wielomianu z x^1 , analogicznie dla każdego indeksu aż do n – $A[n]$ to element, którego x jest podniesiony do n -tej potęgi.

Dla przykładu tablica $A[] = \{5, 0, 1\}$ to wielomian $W(x) = 5 + x^2$
 $B[] = \{3, 2, 0, 5\}$ to wielomian $W(x) = 3 + 2x + 5x^3$

Aby wywołać funkcję dodającą wielomiany musimy ją wywołać z poszczególnymi argumentami:

$A[]$ - Pierwsza tablica

$B[]$ – Druga tablica

n – wielkość tablicy A

m – wielkość tablicy B

np. Dodawanie(A, B, n, m);

Aby wywołać funkcję odejmującą wielomiany musimy ją wywołać z poszczególnymi argumentami:

$A[]$ - Pierwsza tablica

$B[]$ – Druga tablica

n – wielkość tablicy A

m – wielkość tablicy B

np. Odejmowanie(A, B, n, m);

Aby wywołać funkcję mnożącą wielomiany musimy ją wywołać z poszczególnymi argumentami:

$A[]$ – Pierwsza tablica

$B[]$ – Druga tablica

n – wielkość tablicy A

m – wielkość tablicy B

np. Mnozenie(A, B, n, m);

Aby wywołać funkcję wykorzystującą algorytm Hornera do policzenia wartości wielomianu dla podanego przez nas x musimy wywołać ją z argumentami:

$A[]$ – wielomian którego wartość chcemy obliczyć

n – stopień wielomianu

x – podana przez nas wartość x dla której chcemy wyliczyć wartość wielomianu $A[]$

np. Horner($A, 2, 2$)