

ONP – zastosowanie stosu

Sprawozdanie z laboratorium 3 – Piotr Sarna LK1

Cel ćwiczenia

Podczas zajęć zapoznaliśmy się z dwoma algorytmami. Pierwszy z nich miał za zadanie wyznaczenie wartości wyrażenia zapisanego w postaci ONP (Odwrotnej Notacji Polskiej). Drugi z nich miał za zadanie przekształcić wyrażenie zapisane w postaci infiksowej (normalnej) na postać ONP. Trzeci łączył funkcjonalności dwóch poprzednich. Podczas tego ćwiczenia zapoznaliśmy się z operacjami na stosie w C++.

Wstęp teoretyczny

ONP – Odwrotna Notacja Polska to sposób zapisywania wyrażeń matematycznych, pozwalający na zapis bez używania nawiasów, przy jednoczesnym zachowaniu odpowiedniej kolejności wykonywania działań.

Stos – struktura danych typu LIFO (last in first out) wykorzystywana przy obliczaniu wyrażeń zapisanych w ONP. Elementy w niej układamy kolejno jeden na drugim, a dostęp mamy jedynie do ostatnio dodanego elementu, znajdującego się na szczycie stosu. Aby uzyskać dostęp do któregoś z wcześniej dodanych elementów musimy najpierw zdjąć ze stosu te dodane najwcześniej, ponieważ znajdują się one na jego szczycie i blokują nam dostęp do pozostałych.

Opis algorytmów

1. Algorytm obliczania wartości wyrażenia w postaci ONP

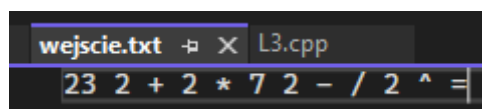
Podane wyrażenie odczytujemy od lewej do prawej, do momentu, aż natrafimy na znak oznaczający koniec równania „=”. Działanie algorytmu przedstawia, dlaczego stos jest idealną strukturą danych do wykorzystania tym przypadku:

- Jeśli odczytaną wartością jest liczba, umieszczamy ją na stosie.
- Jeśli natomiast operator, zdejmujemy ze stosu oczekiwaną przez niego liczbę elementów, wykonujemy działanie matematyczne a następnie jego wynik umieszczamy na stosie.

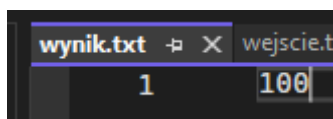
Zapis algorytmu w pseudokodzie:

```
zdefiniuj stos liczbowy  
dopóki odczytany token nie jest '=' powtarzaj  
    jeśli token jest operandem to odłóż go na stos  
    w przeciwnym przypadku (tj. jeśli token jest operatorem) to  
        zdejmij ze stosu odpowiednią liczbę tokenów/operandów (w zależności o operatora)  
        wykonaj obliczenia zgodne z operatorem  
        odłóż wynik obliczeń na stos  
zdejmij z stosu wynik i zwróć go
```

Prezentacja działania mojej implementacji w C++



```
Konsola debugowania programu Microsoft Visual Studio
ONP - zastosowanie stosu
1. Algorytm obliczania wartosci wyrazenia w postaci ONP
2. Algorytm przekształcenia wyrazenia do postaci ONP
3. Algorytm przekształcenia wyrazenia do postaci ONP i obliczenia wartosci wyrazenia w postaci ONP
Wybierz algorytm:
1
1. Pobierz dane z klawiatury
2. Pobierz dane z pliku wejscie.txt
Wybierz opcje:
2
Wynik zostal zapisany do pliku wynik.txt
Czas dzialania algorytmu wynosi 276 mikrosekund
D:\Piotr\Studia\2_semestr\Metody Programowania\L3\x64\Debug\L3.exe (proces 16532) zakończono z kodem 0
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknąć konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```



2. Algorytm przekształcenia wyrażenia do postaci ONP

Aby zamienić wyrażenie z postaci infixowej na ONP wykonujemy następujące kroki, które powtarzamy, aż natrafimy na znak oznaczający koniec równania „=”. W przypadku tego algorytmu również korzystamy ze stosu:

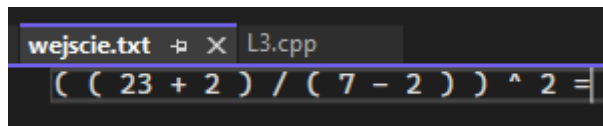
- Jeśli odczytany element jest liczbą, przenosimy ją na wyjście
- Jeśli odczytanym elementem jest nawias otwierający, to umieszczamy go na stosie znaków
- Jeśli odczytanym elementem jest nawias zamykający, to ze stosu przesyłamy na wyjście wszystkie operatory, aż do napotkania nawiasu otwierającego, który usuwamy ze stosu
- Jeśli odczytanym elementem jest operator to dopóki stos jest niepusty oraz odczytany operator ma łączność lewostronną oraz priorytet niższy lub równy operatorowi na stosie lub odczytany operator ma łączność prawostronną i priorytet niższy od operatora na stosie, to pobieramy ze stosu operator i przesyłamy go na wyjście

Zapis algorytmu w pseudokodzie:

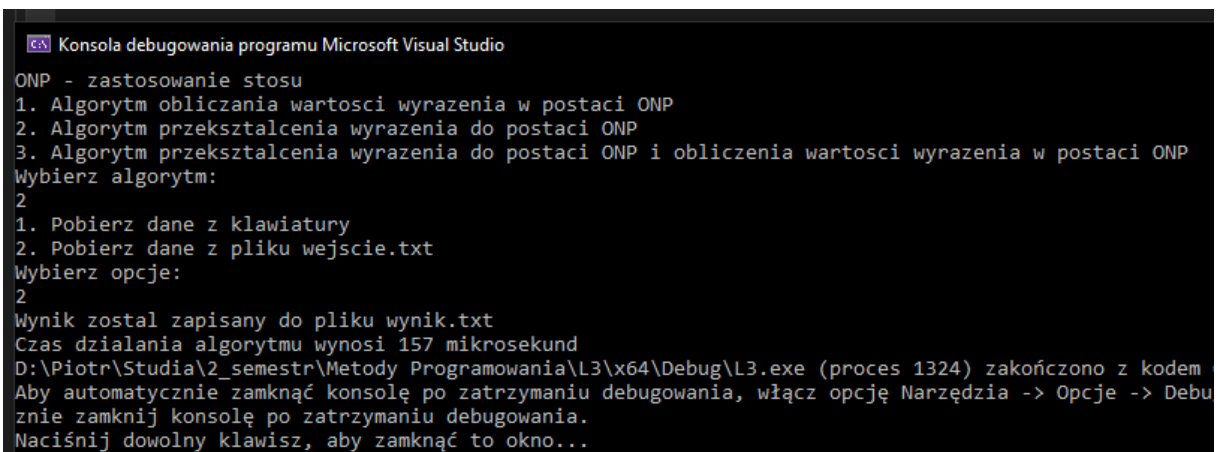
```
zdefiniuj stos tekstowy
zdefiniuj tablicę priorytetów i łączności operatorów

dopóki token nie jest "=" powtarzaj
    jeśli token jest operandem to przenieś go na wyjście
    jeśli token jest ( to odłóż go na stos
    jeśli token jest ) to
        dopóki na stosie jest operator // pętla kończy się po napotkaniu (, który usuwa
            zdejmuj operator ze stosu i przenieś go na wyjście
    jeśli token jest operatorem to
        dopóki na stosie jest operator i token jest operatorem o lewostronnej łączności i i priorytecie niższym lub równym operatorowi na stosie lub operatorem o
        prawostronnej łączności i priorytecie niższym niż operator na stosie
            zdejmij operator ze stosu i przenieś go na wyjście
        odłóż token na stos
```

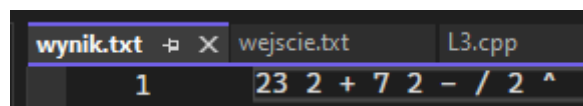
Prezentacja działania mojej implementacji w C++



```
wejście.txt X L3.cpp
( ( 23 + 2 ) / ( 7 - 2 ) ) ^ 2 =
```



```
Konsola debugowania programu Microsoft Visual Studio
ONP - zastosowanie stosu
1. Algorytm obliczania wartości wyrażenia w postaci ONP
2. Algorytm przekształcenia wyrażenia do postaci ONP
3. Algorytm przekształcenia wyrażenia do postaci ONP i obliczenia wartości wyrażenia w postaci ONP
Wybierz algorytm:
2
1. Pobierz dane z klawiatury
2. Pobierz dane z pliku wejście.txt
Wybierz opcje:
2
Wynik został zapisany do pliku wynik.txt
Czas działania algorytmu wynosi 157 mikrosekund
D:\Piotr\Studia\2_semestr\Metody Programowania\L3\x64\Debug\L3.exe (proces 1324) zakończono z kodem
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie
Naciśnij dowolny klawisz, aby zamknąć to okno...
```



```
wynik.txt X wejście.txt L3.cpp
1 23 2 + 7 2 - / 2 ^
```

3. Algorytm przekształcający wyrażenie do postaci ONP i wyliczający jego wartość

Algorytm ten wykorzystuje dwie funkcje przedstawione w poprzednich podpunktach, do funkcji wyliczającej wartość ONP jest przesyłany wynik funkcji zamieniającej wyrażenie na ONP

Wnioski

Powyższe algorytmy wykorzystujące stos cechuje świetna, liniowa złożoność obliczeniowa $O(n)$, dzięki czemu wykorzystywanie ich jest opłacalne pod względem czasowym.

Bibliografia

<https://www.algorytm.edu.pl/algorytmy-maturalne/onp>

<https://stackoverflow.com/questions/5888022/split-string-by-single-spaces>