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 ja 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:

Obraz zawierający tekst, zrzut ekranu, Czcionka, algebra

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Prezentacja działania mojej implementacji w C++

Obraz zawierający tekst, Czcionka, zrzut ekranu, zegar

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

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:

Obraz zawierający tekst, zrzut ekranu, Czcionka, algebra

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Prezentacja działania mojej implementacji w C++

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

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>