Algorytmy i struktury danych
LABORATORIUM

Zajęcia 7
Struktury danych: stos (kontynuacja z poprzednich zajęć), kopiec

Cel zajęć

Zapoznanie studentów ze strukturami danych stosu, kolejki priorytetowej oraz kopca, implementacja wybranych problemów algorytmicznych wykorzystujących tę strukturę.

Zadania

1. Dane jest wyrażenie 3+7*(4-2)+2*(3+8)-1 (dla uproszczenia wykorzystujemy na razie liczby jednocyfrowe). Zaimplementuj algorytm Dijkstry, pozwalający obliczyć wartość tego wyrażenia z wykorzystaniem dwóch stosów: stosu operandów(wartości) i stosu operatorów:
    c. Analizujemy od lewej po jednym elemencie (liczba, operator, nawias, itp.). Gdy napotkamy:
        i. *Wartość:* dodaj ją na stos wartości
        ii. *Operator:* dodaj go na stos operatorów
        iii. *Lewy nawias:* ignoruj go
        iv. *Prawy nawias:* zdejmij operator i dwie wartości z odpowiednich stosów; połóż rezultat zastosowania operatora do tych wartości na stos wartości
    d. Jeśli badane wyrażenie nie zostało wyczerpane – powtarzamy a., a jeśli tak - wartość znajdująca się na stosie wartości jest wynikiem końcowym.
2. Dane jest wyrażenie jw. Zaimplementuj algorytm J. Łukasiewicza, wykorzystujący jeden stos wspólny dla wartości oraz operatorów.
    a. Przekształć wyrażenie do notacji postfiksowej (odwrotna notacja polska ONP – ang. *Polish reverse notation*). Skorzystaj z: https://raj457036.github.io/Simple-Tools/prefixAndPostfixConvertor.html
    b. Analizujemy od lewej po jednym elemencie (liczba, ogranicznik, itp.). Gdy napotkamy:
        i. *Wartość*: dodaj ją na stos wartości
        ii. *Operator*: zdejmij ze stosu odpowiednią dla danego operatora liczbę argumentów, wykonaj na nich obliczenia, a uzyskany wynik połóż na stos.
    c. Jeśli badane wyrażenie nie zostało wyczerpane – powtarzamy, a jeśli tak - wartość znajdująca się na stosie jest wynikiem końcowym.
3. Napisz swoją funkcję przekształcającą wyrażenie z notacji infiksowej na postfiksową.
4. Wykonaj powyższe zadania, zakładając możliwość wystąpienia liczb dwu i więcej cyfrowych w wyrażeniu.
5. Jak zachowają się powyższe programy, jeżeli wyrażenie będzie niepoprawnie skonstruowane, np. brak równowagi nawiasów, dwa operatory występujące obok siebie, itp.?
6. Napisz funkcję, sprawdzającą, czy dana tablica liczb całkowitych jest kopcem: `boolean isHeap(int[] T)`. Dla danej tablicy T=[0, 12, 3, 7, 11, 10, 8, 2, 10, 9, 15] sprawdź, czy reprezentuje ona kopiec. Wykorzystaj funkcję `isHeap()`. Wydrukuj stosowny komunikat.
7. Przedstaw kopiec uzyskany po wstawieniu znaków: ALFAROMEO (w tej kolejności) do początkowo pustego kopca z łatwym dostępem do maksimum.