Temat: Struktury dynamiczne do realizacji algorytmu: ONP i symulacji problemu Flawiusza

1. Odwrotna Notacja Polska (ONP)

Wprowadzenie

Odwrotna Notacja Polska (ONP) to zapis wyrażeń matematycznych, w którym operatory umieszczane są za argumentami. Przykład wyrażenia:

- Klasyczna notacja: (3 + 4) * 5
- ONP: 3 4 + 5 *

Aby obliczyć wynik wyrażenia zapisanego w ONP, wykorzystujemy stos:

1. Przechodzimy przez wyrażenie od lewej do prawej.
2. Liczby umieszczamy na stosie.
3. Po napotkaniu operatora, pobieramy dwie liczby ze stosu, wykonujemy działanie i wynik odkładamy na stos.
4. Po przejściu całego wyrażenia na stosie pozostaje jedna liczba – wynik.

Algorytm ONP

1. Utwórz pusty stos.
2. Przechodź przez każdy element wyrażenia:
    - Jeśli element to liczba, dodaj do wyniku.
    - Jeśli element to operator, usuwaj ze stosu elementy zgodnie z priorytetem i dodawaj do wyniku, a operator umieść na stosie.
3. Po przejściu wyrażenia przenieś pozostałe operatory ze stosu do wyniku.

Kod w Pythonie

```
def onp(wyrazenie):
    stos = []
    for token in wyrazenie.split():
        if token.isdigit():
            stos.append(int(token))
        else:
            b = stos.pop()
            a = stos.pop()
            if token == '+':
                stos.append(a + b)
            elif token == '-':
                stos.append(a - b)
            elif token == '*':
                stos.append(a * b)
            elif token == '/':
                stos.append(a / b)
    return stos.pop()

wyrazenie = "3 4 + 5 *"
print("Wynik ONP:", onp(wyrazenie))
```

2. Problem Flawiusza

Wprowadzenie

Problem Flawiusza polega na symulacji eliminacji osób w okręgu według zadanej reguły. Przykład:

- N osób ustawionych w okręg numerujemy od 1 do N.
- Co k-tą osobę usuwamy, a proces kontynuujemy, aż zostanie jedna osoba.
- Struktura dynamiczna, np. lista cykliczna, pozwala na efektywną symulację tego procesu.

Algorytm problemu Flawiusza

1. Utwórz listę osób od 1 do N.

2. Ustaw licznik na 0.
3. Iteracyjnie przesuwaj się o k-1 pozycji, usuwając osobę.
4. Kontynuuj, aż zostanie jedna osoba.

Kod w Pythonie

```
def flawiusz (n, k):
    osoby = list(range(1, n + 1))
    idx = 0
    while len(osoby) > 1:
        idx = (idx + k - 1) % len(osoby)
        osoby.pop(idx)
    return osoby[0]

n = 7  # liczba osób
k = 3  # co k-ta osoba jest usuwana
print("Ocalona osoba:", flawiusz(n, k))
```

ZADANIA DO WYKONANIA

ZAD.1 Oblicz wynik wyrażenia zapisanego w ONP: `2 3 + 4 *`.

ZAD.2 Rozwiąż problem Flawiusza dla N = 5 i k = 2.

ZAD.3 Zaimplementuj algorytm konwersji wyrażeń infiksowych do ONP.

- **Podpowiedź:** W wyrażeniu infiksowym kolejność operacji jest determinowana przez priorytety operatorów i nawiasy. Przykład:
  - Wyrażenie infiksowe: `(3 + 4) * 5`
  - Krok 1: Operator `+` ma niższy priorytet niż `*`, ale nawias wymusza jego pierwszeństwo.
  - Krok 2: Konwersja: `3 4 + 5 *` (ONP).

ZAD.4 Rozwiąż problem Flawiusza, wypisując w kolejności osoby usuwane dla N = 10 i k = 4.

ZAD.5 Rozszerz problem Flawiusza, aby co k-tą osobę eliminować w pierwszej rundzie, a następnie co m-tą w kolejnych rundach