



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA

Karolina Magdoń

Inżynieria i analiza danych
I rok

Sprawozdanie z projektu przesuwania elementów tablicy (C++)

Praca przygotowana na zajęcia
„Algorytmy i struktury danych”
w roku akademickim 2022/2023

Rzeszów, listopad 2022

Spis treści

1. Wstęp i opis zagadnień projektu	3
2. Podstawy teoretyczne zagadnienia	3
4. Opis algorytmu przesuwającego	4
5. Złożoność obliczeniowa	4
6. Schemat blokowy	5
7. Pseudokod	6
9. Podsumowanie i wnioski	7

1. Wstęp i opis zagadnień projektu

Moim zadaniem projektowym było utworzenie programu, który dla zadanej tablicy liczb całkowitych przesunie wszystkie elementy mniejsze od 0 na jej koniec. Dodatkowym warunkiem było zachowanie kolejności występowania elementów ujemnych. Do kolejnych zaleceń należało: przedstawienie podstaw teoretycznych zagadnienia oraz schemat blokowy algorytmu (również pseudokod). Zobrazować rezultaty działania programu. Ukazać złożoność czasową, obliczeniową.

2. Podstawy teoretyczne zagadnienia

3. Cechy programu

(.....1.Możliwość odczytywania danych wejściowych z pliku tekstowego i zapisu posortowanego ciągu do pliku tekstowego z danymi wyjściowymi.

2.Na potrzeby testów zaimplementowanie funkcji generującej „losowe” ciągi elementów (o zadanej długości) i zapisującą je do pliku tekstowego z wyjściowymi danymi.

3.Założenie, że posortowane elementy to liczby całkowite z przedziału $[0, N]$, gdzie N powinno być „odpowiednio dużym” parametrem wewnątrz programu.

4.Kod opatrzony stosownymi komentarzami.....)

Do wykonania tego projektu stworzyłam kod w języku C++ zawierający:

- Funkcję generującą losowy zbiór liczb służący do testowania algorytmu przesuwanego
- Właściwy algorytm przesuwanący

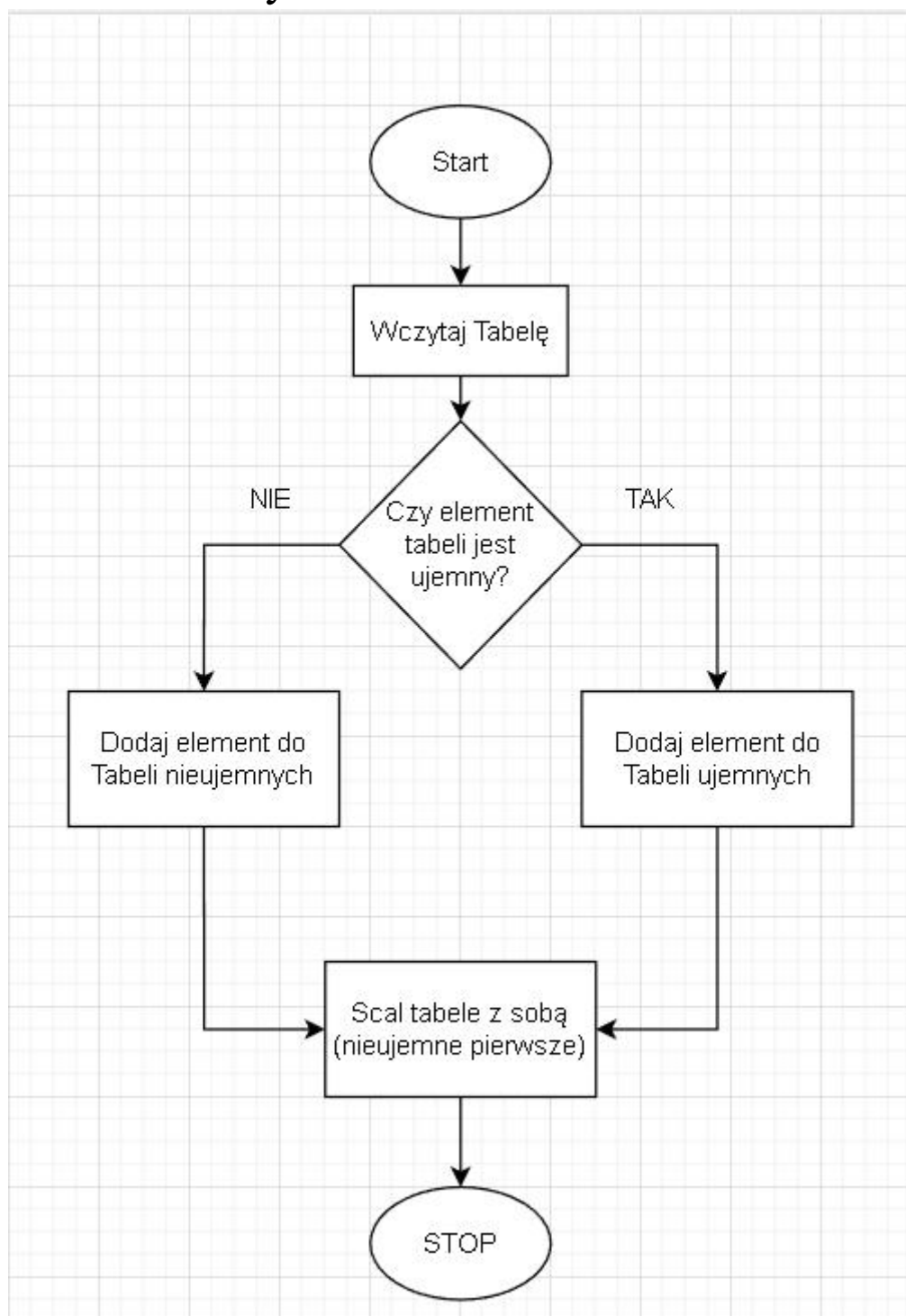
- Zaimplementowany do algorytmu przesuwającego miernik czasu działania
- (może coś o tej funkcji już wypisującej wynik?)

4. Opis algorytmu przesuwającego

5. Złożoność obliczeniowa

Złożoność obliczeniowa algorytmu jest bardzo wydajna, w każdym wypadku wynosi $O(n)$, gdyż wystarczy, że tylko 2 razy przeanalizujemy wszystkie liczby z tabeli, aby wyznaczyć wynik. Złożoność czasowa wynosi *dopisane po zrobieniu pomiaru czasu*.

6. Schemat blokowy



7. Pseudokod

K01: Wczytaj Tabelę

K02: Stwórz 2 Tabele Pomocnicze:

1. Tabela która będzie przetrzymywać liczby nieujemne
2. Tabela która będzie przetrzymywać liczby ujemne

K03: Stwórz 2 Zmienne Pomocnicze:

1. Zmienna która będzie wskaźnikiem tabeli z liczbami nieujemnymi
2. Zmienna która będzie wskaźnikiem tabeli z liczbami ujemnymi

K04: Patrzymy na każdy element po kolei i sprawdzamy dla niego, czy jest mniejszy od 0, jeśli nie, przechodzimy do K04.1, jeśli tak, do K04.2

K04.1 Dopisz element do Tabeli dla elementów nieujemnych na miejsce, które wskazuje wskaźnik i zwiększ wartość wskaźnika tabeli z liczbami nieujemnymi

K04.2 Dopisz element do Tabeli dla elementów ujemnych na miejsce, które wskazuje wskaźnik i zwiększ wartość wskaźnika tabeli z liczbami ujemnymi

K05: Scalamy Tabele ze sobą (nieujemne jako pierwsza)

K06: Wypisz Tabelę.

8. Rezultaty testów

Algorytm w każdym przypadku poprawnie wykonał przesuwanie na Tabeli. Oto parę przykładów:

```
Wygenerowana Tablica = [ 6 -7 8 4 -1 -7 9 8 -3 9 ]  
Tablica po przesunięciu = [ 6 8 4 9 8 9 -7 -1 -7 -3 ]
```

```
Wygenerowana Tablica = [ -7 6 -3 -3 -6 8 4 9 8 -3 ]  
Tablica po przesunięciu = [ 6 8 4 9 8 -7 -3 -3 -6 -3 ]
```

```
Wygenerowana Tablica = [ -2 6 -8 3 -2 -1 -2 0 0 8 ]  
Tablica po przesunięciu = [ 6 3 0 0 8 -2 -8 -2 -1 -2 ]
```

```
Wygenerowana Tablica = [ 3 7 9 -3 4 2 -8 -9 3 -2 1  
Tablica po przesunięciu = [ 3 7 9 4 2 3 -3 -8 -9 -2 1
```

```
Wygenerowana Tablica = [ -5 -4 5 2 9 -6 -10 7 2 -8 1  
Tablica po przesunięciu = [ 5 2 9 7 2 -5 -4 -6 -10 -8 1
```

Tutaj też wykresy zmierzonego czasu działania zależnie od ilości liczb.

9. Podsumowanie i wnioski

Założenia projektu udało się zrealizować relatywnie nieskomplikowanym obliczeniowo algorytmem, ponadto poradził on sobie *tu zależnie od tego jak sobie poradził w testach czasowych*. Przez specyfikę zagadnienia użycie algorytmu sortującego nie dawało nam pożądanego wyniku, więc użyłam w moim algorytmie bardziej codziennego znaczenia słowa sortowanie i podzieliłam liczby na ujemne i nieujemne. Ponieważ przechodziliśmy po tabelach w kolejności od lewej do prawej, tabele po „sortowaniu” mogą zostać ze sobą złożone bez dodatkowych operacji.