

Nikodem Krupa

Algorytmy i struktury danych

P04

Opis problemu

Trzeba znaleźć wszystkie podciągi dwuelementowe ,w których zera poprzedza ta sama liczba jedynek.

Przykład:

Wejście [0 1 1 1 1 0 0 0 1 0 1 1 0 0]

Wyjście[1 1 1 0 0 0]

Wejście [0 1 0 0 0 1 0 0 1 0 1]

Wyjście brak elementów spełniających dane zadania.

Funkcje użyte w programie

1. Generuj_dane

- generuje losowe liczby z przedziału 0 i 1

2. Generuj_Jedynki _przed_zerami

- na podstawie warunków wypisuje podciągi większe niż dwuelementowe których zera poprzedza taka sama liczba jedynek i wypisuje je do tablicy.

Opis działania Programu

Program ma za zadanie wpisanie do pliku wynik.txt losowych wygenerowanych liczb 0 i 1 w ilości podanej przez użytkownika. Następnie funkcja znajduje podciągi które spełniają opisane warunki w funkcji. Oraz również wpisuje je do pliku wynik.txt. istnieje możliwość testów programu w main.cpp.

Schematy blokowe

Podam poniżej dwa schematy blokowe jeden funkcji generowania podciągów dwu i więcej elementowych a drugi ilość testów z każdym kolejnym ciągiem większym o 5 elementów .

Pseudokod

Dla każdego elementu tablicy tab

Jeśli $\text{tab}[j]=1$ i $\text{tab}[j-1]=1$ i $j \geq 1$ to

$s0 \leftarrow 0$

$s1 \leftarrow 0$

w przeciwnym razie jeśli $\text{tab}[j]=1$ to

$s1 \leftarrow s1 + 1$

w przeciwnym razie jeśli $s1 \geq 2$ to

$s0 \leftarrow s + 1$

jeśli $s0 \geq 2$ i $s1 \geq 2$ i $\text{tab}[j+1]=1$

$\text{dane} = \text{dane} + "["$

$m \leftarrow \text{minimum}(s1, s0)$

dla $p=0$ dopoki $p < m$ to $p+1$

$\text{dane} \leftarrow \text{dane} + "1"$

dla $p=0$ dopoki $p < m$ to $p+1$

$\text{dane} = \text{dane} + "0"$

$\text{dane} = \text{dane} + "]"$

jeśli $s0 \geq 2$ i $s1 \geq 2$

$\text{dane} = \text{dane} + "["$

$m \leftarrow \text{minimum}(s1, s0)$

dla $p=0$ dopoki $p < m$ to $p+1$

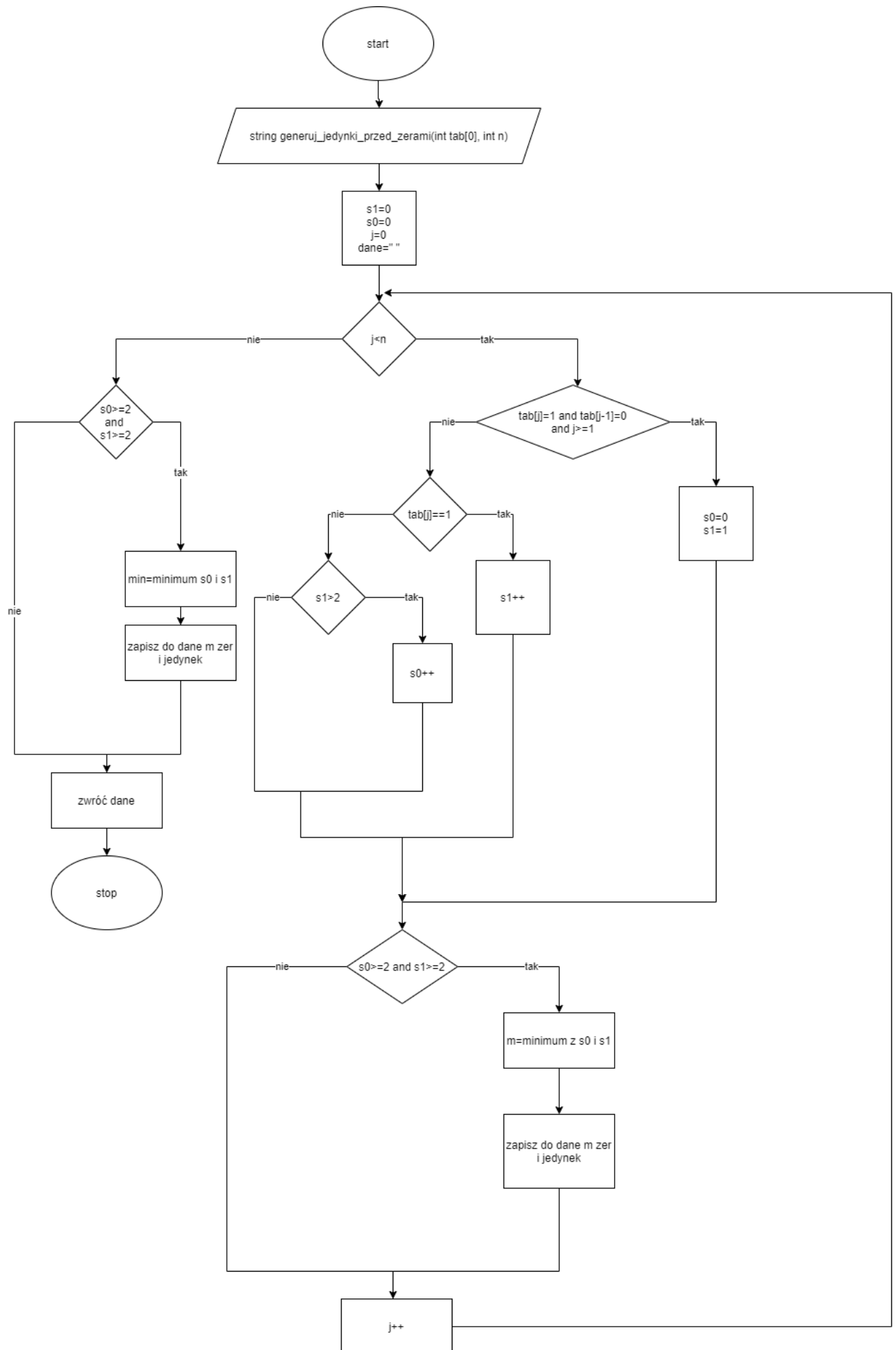
$\text{dane} \leftarrow \text{dane} + "1"$

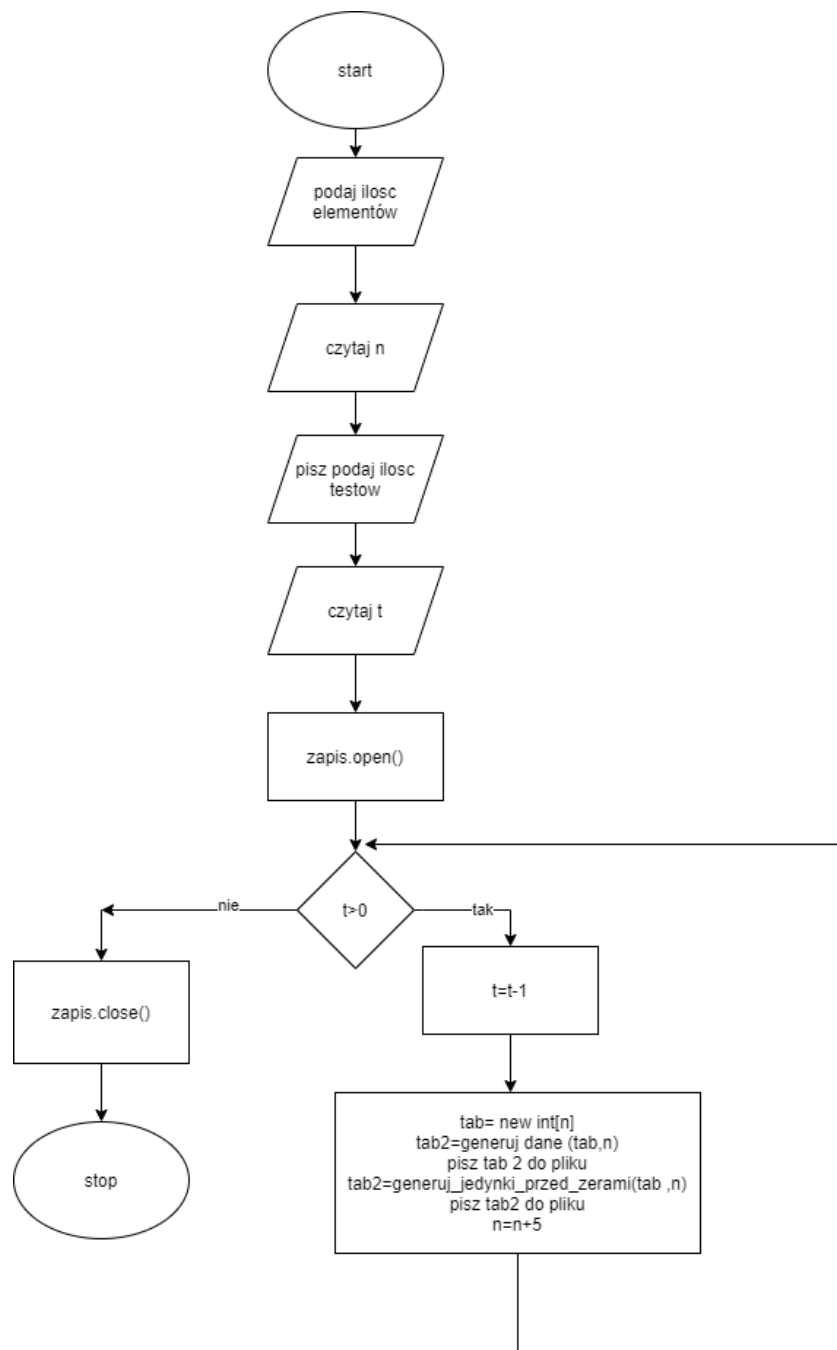
dla $p=0$ dopoki $p < m$ to $p+1$

$\text{dane} = \text{dane} + "0"$

$\text{dane} = \text{dane} + "]"$

zwróć dane





Czas działania algorytmu w mikrosekundach

Dla 10 elementów	558 mikrosekund
Dla 100 elementów	839 mikrosekund
Dla 1000 elementów	1497 mikrosekund
Dla 10000 elementów	50079 mikrosekund

Czas działania wykres w mikrosekundach



Wnioski:

Opisany algorytm poprawnie zlicza podciągi jedynek i zer

```
wyniki.txt — Notatnik
Plik Edycja Format Widok Pomoc
[1 0 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0]
[1 1 0 0 ]
[1 1 0 0 ]
[1 1 0 0 ]
[1 1 0 0 ]
[1 1 0 0 ]
[1 1 1 0 0 0 ]

C:\Users\Nikodem\Desktop\tabcice\bin\Debug\tabcice.exe
podaj ilosc elementow w tablicy skladajacej sie z zer i jedynek
100
Podaj ilosc testow: 1
Wyniki zapisano do pliku Wyniki.txt
Process returned 0 (0x0)   execution time : 3.834 s
Press any key to continue.

wyniki.txt — Notatnik
Plik Edycja Format Widok Pomoc
[0 1 1 1 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 0 1 0 1 1 1 0 0 0 0 1 0]
[1 1 0 0 ]
[1 1 1 0 0 0 ]
[1 1 1 0 0 0 ]
[1 1 0 0 ]
[1 1 1 0 0 0 ]
[1 1 0 0 ]
[1 1 0 0 ]

C:\Users\Nikodem\Desktop\tabcice\bin\Debug\tabcice.exe
podaj ilosc elementow w tablicy skladajacej sie z zer i jedynek
95
Podaj ilosc testow: 2
Wyniki zapisano do pliku Wyniki.txt
Process returned 0 (0x0)   execution time : 3.297 s
Press any key to continue.
```