

Raport z realizacji mini-projektu

Numer projektu: 3

Autor: Ferkaluk Antoni

Numer albumu: 22606

Kod grupy zajęciowej: IZ01P06

1. Treść zadania.

3. Dane są dwa zbiory A i B (n-elementowy i m-elementowy) o elementach będących liczbami naturalnymi. Elementy tych zbiorów dane są za pomocą dwóch tablic uporządkowanych rosnąco. Skonstruuuj algorytm wyznaczania liczby elementów o wartościach nieparzystych należących do zbioru $A \oplus B$ (suma rozłączna zbiorów).

Np. $A=\{1, 2, 3, 5, 8\}$, $B=\{0, 1, 3, 4, 8, 9, 10\}$,

$A \oplus B =\{0, 2, 4, 5, 9, 10\}$, ale wartości nieparzyste to: 5, 9.

Liczba szukanych elementów to 2.

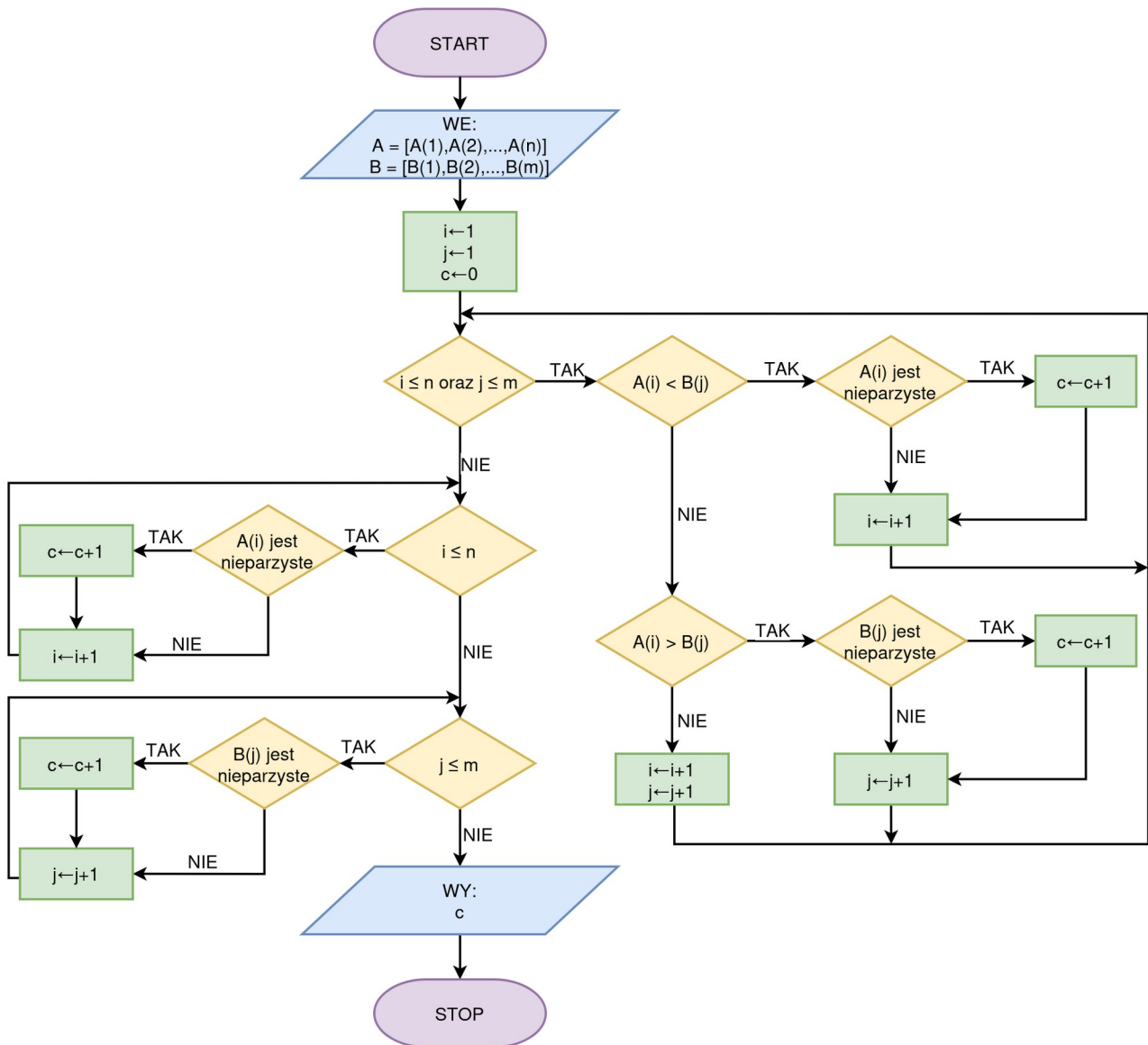
2. Opis słowny algorytmu

Dane wejściowe: dwie kolejno n-elementowe i m-elementowe posortowane rosnąco tablice, zawierające liczby naturalne.

Oczekiwane dane wyjściowe: ilość elementów będących liczbami nieparzystymi, powstałych w wyniku operacji sumy rozłącznej na danych zbiorach.

1. Wprowadź zmienną, w której przechowany będzie wynik końcowy
2. wskaż pierwsze elementy zbiorów A i B
3. wykonuj, dopóki, któryś ze wskaźników nie wskaże elementu spoza jego zbioru:
 - 3.1. porównaj wskazane elementy
 - 3.2. jeżeli wskazany element ze zbioru A jest mniejszy od wskazanego elementu ze zbioru B, wskaż kolejny element ze zbioru A, oraz inkrementuj zmienną wyniku końcowego, jeżeli sprawdzany element ze zbioru A jest nieparzysty.
 - 3.3. jeżeli wskazany element ze zbioru A jest większy od wskazanego elementu ze zbioru B, wskaż kolejny element ze zbioru B, oraz inkrementuj zmienną wyniku końcowego, jeżeli sprawdzany element ze zbioru B jest nieparzysty.
 - 3.4. jeżeli wskazane elementy mają taką samą wartość, wskaż kolejne elementy z obu zbiorów.
4. wykonuj aż do wskazania ostatniego elementu z A:
 - 4.1. inkrementuj zmienną wyniku końcowego, jeżeli sprawdzany element ze zbioru A jest nieparzysty.
 - 4.2. wskaż kolejny element ze zbioru A.
5. wykonuj aż do wskazania ostatniego elementu z B:
 - 5.1. inkrementuj zmienną wyniku końcowego, jeżeli sprawdzany element ze zbioru B jest nieparzysty.
 - 5.2. wskaż kolejny element ze zbioru B
6. zwróć zmienną wyniku końcowego.

3. Schemat blokowy



4. Symulacja działania dla przykładowych danych

A	B	i	j	c	i ≤ n oraz j ≤ m	i ≤ n	j ≤ m	A(i) < B(j)	A(i) > B(j)	A(i) nieparzyste	B(j) nieparzyste
[4,5,6]	[1,2,6,8,9]	1	1	0	TAK			NIE	TAK		TAK
			2	1	TAK			NIE	TAK		NIE
			3		TAK			TAK		NIE	
		2			TAK			NIE		TAK	
		3		2	TAK			NIE	NIE		
		4	4		NIE	NIE	TAK				NIE
			5				TAK				TAK
			6	3			NIE				

5. Zapis algorytmu w języku Python

```
1  def disjointUnionOddCount( A, B ):
2      n, m = len( A ), len( B )
3      i, j = 0, 0
4      c = 0
5
6      while i < n and j < m:
7          if A[ i ] < B[ j ]:          # A[ i ] nie występuje w B
8              if A[ i ] & 1 == 1:      # oraz jest nieparzyste
9                  c += 1
10                 i += 1
11
12             elif A[ i ] > B[ j ]:     # B[ j ] nie występuje w A
13                 if B[ j ] & 1 == 1:   # oraz jest nieparzyste
14                     c += 1
15                     j += 1
16
17             else:                    # A[ i ] == B[ j ] pomini
18                 i += 1
19                 j += 1
20
21             # W przypadku gdy jedna tablica jest większa od drugiej
22             # sprawdź pozostałe elementy
23
24         while i < n:
25             if A[ i ] & 1 == 1:
26                 c += 1
27                 i += 1
28
29         while j < m:
30             if B[ j ] & 1 == 1:
31                 c += 1
32                 j += 1
33
34     return c
```

6. Oszacowanie złożoności czasowej

a. n, m - rozmiar zadania: kolejno liczba elementów list A i B

b.

1 pętla - Operacja dominująca:

w najgorszym wypadku dwa porównania:

porównanie $A(i) < B(j)$,

porównanie $A(i) > B(j)$

2. pętla - Operacja dominująca:

porównanie $A(i)$ jest nieparzyste

3. pętla - Operacja dominująca:

porównanie $B(j)$ jest nieparzyste

c. Funkcja złożoności:

Opcja 1 ($n < m$):

$$F(n, m) = 2n + 0 + (m - n) = n + m$$

Opcja 2 ($n > m$):

$$F(n, m) = 2m + (n - m) + 0 = n + m$$

Opcja 3 ($n = m$):

$$F(n, m) = n + 0 + 0 = n = m$$

Najgorszy wypadek:

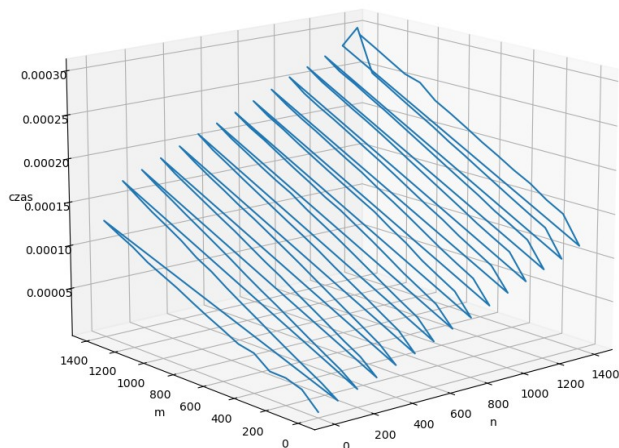
$$F(n, m) = n + m$$

d. Rząd złożoności w najgorszym wypadku:

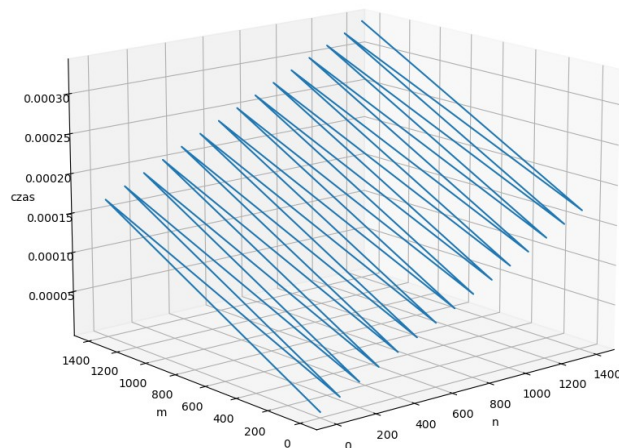
$$F(n, m) = O(n + m) - \text{złożoność liniowa}$$

7. Wykres zależności czasu od rozmiaru zadania (n, m)

Wykres funkcji na podstawie danych losowych



Wykres $n+m$



Wykres został przedstawiony na podstawie losowo wygenerowanych danych, spełniających warunek zadania. Przedstawiony czas jest średnią 350 prób z każdego kroku.

Wniosek:

Porównanie wykresów zależności czasu wykonania funkcji oraz przewidzianej złożoności potwierdza złożoność liniową algorytmu.

8. Podsumowanie

Algorytm sprawnie wyznacza liczbę nieparzystych elementów sumy rozłącznej dwóch zbiorów, osiągając złożoność liniową. Dodatkowo, algorytm nie alokuje dodatkowej pamięci (pomijając iteratory i wynik).