Zadanie projektowe numer 40

Autor: Piotr Świder

# Pseudokod

## Pseudokod funkcji get\_submatrix

Funkcja get\_submatrix(matrix,x,y)

result\_submatrix <- pusta lista dwuwymiarowa

iter\_x <- 0

iter\_y <- 0

Dopóki iter\_y<2

Dopóki iter\_x<2

result\_submatrix[iter\_x][iter\_y] <- matrix[x+iter\_x][y+iter\_y]

iter\_x <- iter\_x +1

iter\_y <- iter\_y +1

iter\_x <- 0

Zwróć result\_submatrix

## Pseudokod funkcji get\_matrix\_sum

Funkcja get\_matrix\_sum(submatrix)

Suma <- 0

Iter\_x <- 0

Iter\_y <- 0

Dopóki Iter\_y<2

Dopóki Iter\_x<2

suma <- suma + submatrix[iter\_x][iter\_y]

iter\_x <- iter\_x + 1

iter\_y < iter\_y + 1

iter\_x <- 0

Zwróć suma

## Finalny pseudokod

Funkcja find\_submatrices\_summing\_above\_threshhold(matrix)

Results <- pusta lista dwuwymiarowa

k <- 25

Dopóki y<długości matrix- 1

Dopóki x<szerokości matrix-1

Submatrix = get.submatrix(matrix,x,y)

Jeśli get\_matrix\_sum(submatrix)>=k

Dodaj submatrix do results

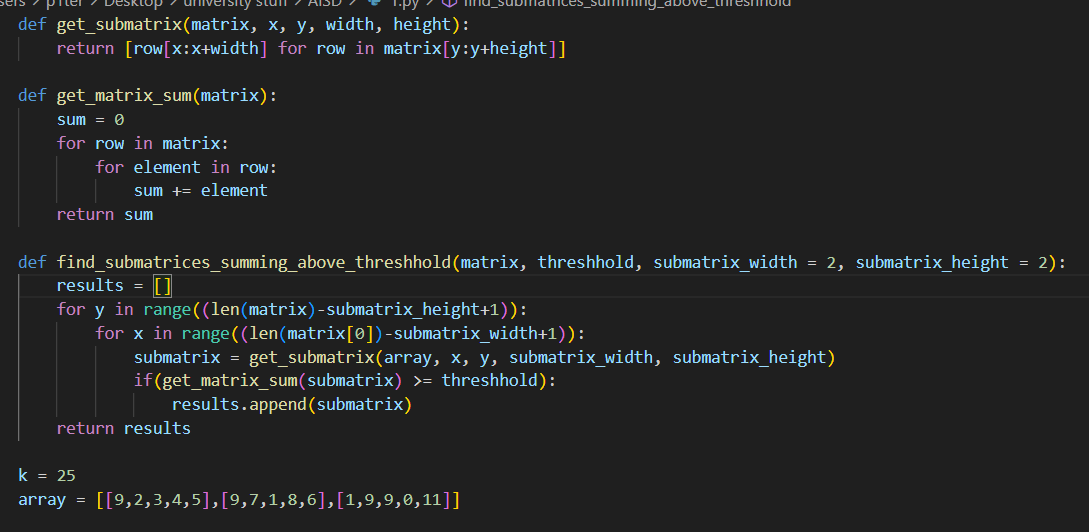
x <- x + 1

y <- y + 1

x <- 0

Zwróć result

# Szacowana złożoność obliczeniowa algorytmu



Zakładając, że wprowadzona została macierz o wymiarach MxN, określamy złożoność obliczeniową w sposób następujący.

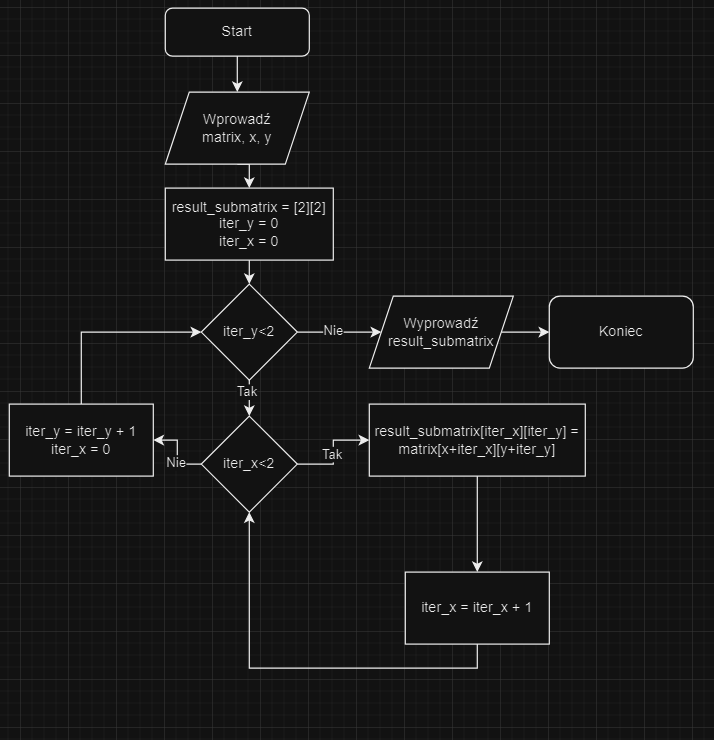
Dla funkcji get\_submatrix złożoność obliczeniowa wyniesie width \* height, przy czym zakładamy, że dla rozpatrywanego przypadku ich wartości zawsze wynoszą 2, więc złożoność wynosi 4.

Dla funkcji get\_matrix\_sum złożoność obliczeniowa wynosi długość \* szerokość zadanej macierzy, przy czym zakładamy, że dla rozpatrywanego przypadku ich wartości zawsze wynoszą 2, więc złożoność, podobnie jak w poprzednim przypadku wynosi 4.

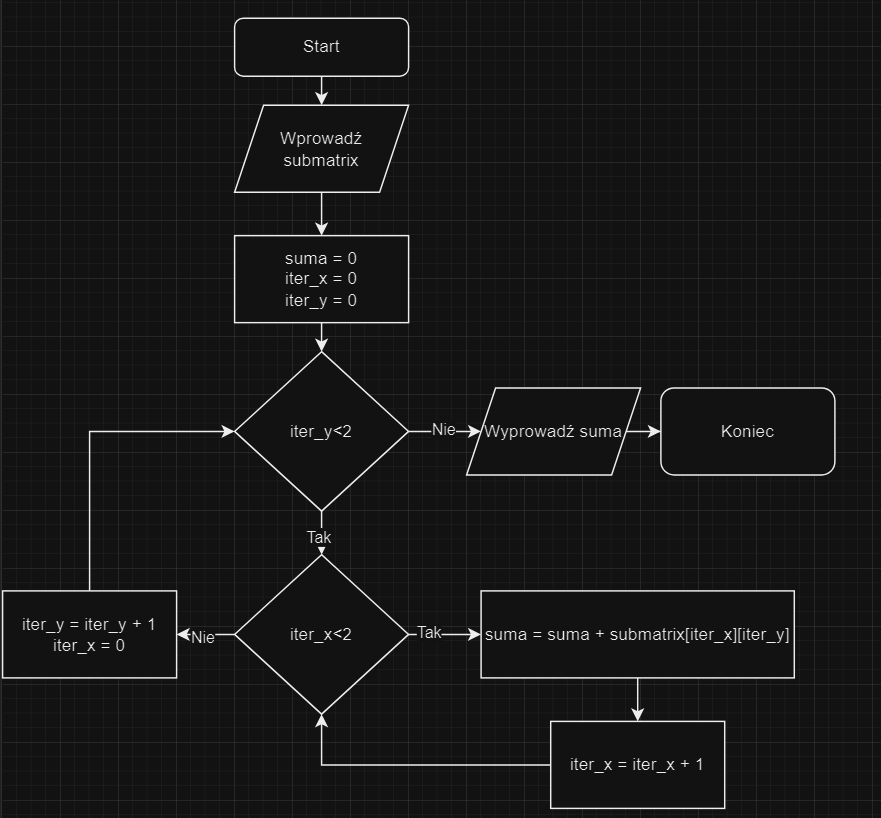
Dla głównej funkcji find\_submatrices\_summing\_above\_threshhold złożoność obliczeniowa wynosi ilość wykonań pierwszej pętli for, która wynosi N pomnożona przez ilość wykonań drugiej pętli for, która wynosi M, które to mnożymy przez ilość operacji wykonanych wewnątrz tych pętli, a więc 4+4. Po wykonaniu wszystkich operacji złożoność wynosi 8\*M\*N, co dalej upraszczamy do O(M\*N).

# Schemat blokowy

## Schemat blokowy funkcji get\_submatrix



## Schemat blokowy funkcji get\_matrix\_sum



## Finalny Schemat blokowy

## 

### Wnioski po ułożeniu schematów blokowych

Jednym z wygodniejszych sposobów, aby opisanie finalnego schematu blokowego było najefektowniejsze było w tym przypadku podzielenie tych schematów na funkcję, aby żadne z wyrażeń nie było zbyt zagnieżdżone i przekombinowane.

W pierwszej kolejności, aby dostać się powoli do finalnego schematu blokowego rozpisujemy działanie funkcji get\_submatrix, która ma za zadanie wyciągnąć z macierzy o rozmiarze MxN (array\_length), który jest wprowadzony przez użytkownika – podmacierz 2x2 tak, aby zrealizować cel zadania.

Drugą funkcją jest get\_matrix\_sum, która w późniejszym etapie jest wykorzystywana do sumowania wartości z funkcji get\_submatrix, która wydobywa nam właśnie podmacierze.

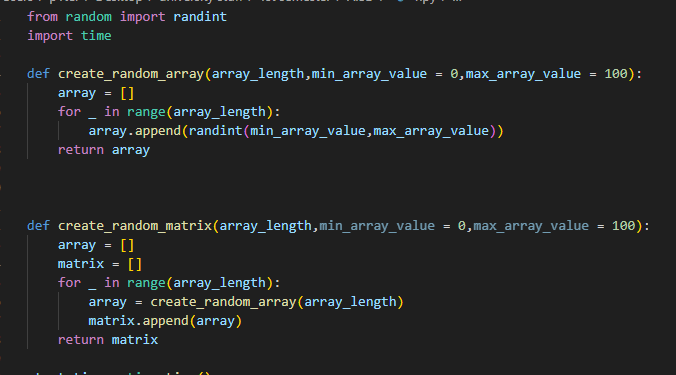
Ostateczną – finalną funkcją jest funkcja find\_submatrices\_above\_threshhold, w której wykorzystujemy obie wcześniej wymienione oraz opisane funkcję, aby sprawdzić czy suma (którą określiliśmy przy pomocy funkcji get\_matrix\_sum) podmaciery wydobyta przez funkcję get\_submatrix jest większa niż nasz threshhold czyli inaczej k, a w naszym przypadku jest to po prostu 25, a na koniec jeśli jest ona większa wypisania jej w tablicy results.

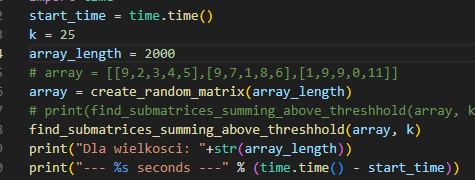
# Czasy obliczeń dla algorytmu w zależności od liczby elementów

# 

Szybkość wzrostu czasu wykonania algorytmu jest zależna liniowo od ilości elementów macierzy co pokazuje powyższy wykres.

Aby stworzyć powyższy wykres wystarczy zastosować funkcję wyliczania czasu na podstawie wielkości macierzy. W tym wypadku stworzyliśmy dwie dodatkowe funkcję, w której jedna tworzy tablicę o wartościach pseudolosowych z zakresu od 0 do 100 o długości array\_length – w tym wypadku zależnie od testowanych danych.



Musieliśmy również zaimportować bibliotekę time, aby złożoność czasowa była możliwa do obliczenia przez program: 

Przykładowe wyniki dla przykładowych wielkości macierzy (a w zasadzie tablic, które tworzą tablicę dwuwymiarową):

