

# **SPRAWOZDANIE Z ZADANIA PROJEKTOWEGO**

Patryk Huk

P03

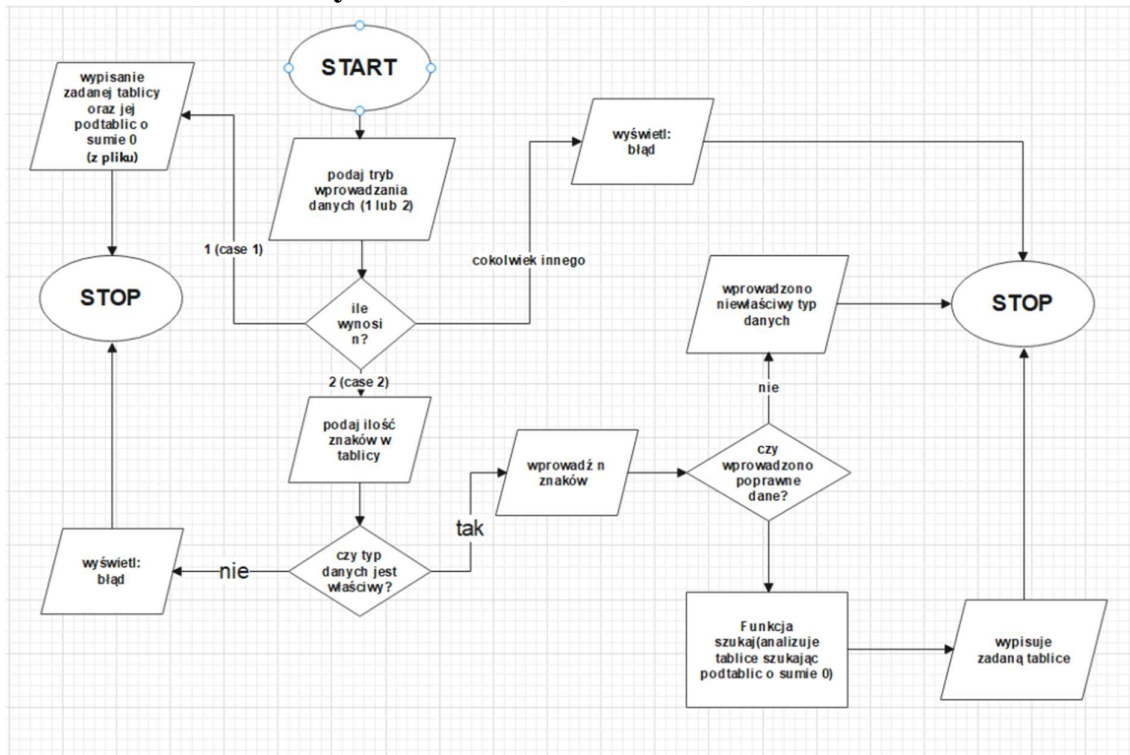
Inżynieria i analiza danych

Szukanie podtablic o sumie 0 dla zadanej tablicy

# 1. Opis problemu

Zadaniem programu jest znalezienie podtablic o sumie 0 w zadanej tablicy oraz wypisanie ich. Będziemy wprowadzali różne dane testując w ten sposób działanie algorytmu, oraz czas jego wykonywania.

## 2. Schemat blokowy



Rys. 1

## 3. Pseudokod

Funkcja wypisz

```
void wypisz(int T[],int p,int q)
```

```
{    wyświetl "[" ;
```

```
dla(int i=p; dopóki i<=q-1;i++)
```

```
wyświetl << T[i]<< " , ";
```

```
wyświetl  T[q]<<" ]" przejdź do nowej linii;
```

```
}
```

Funkcja zapisz

```
void zapisz(int T[],int p,int q,fstream &plik)
```

```
{
```

```
zapisz do pliku "[" ;
```

```

dla(int i=p;dopóki i<=q-1;i++)
    zapisz do pliku T[i] " ";
zapisz do pliku T[q]<<" ]" przejdźcie do nowej linii;
}

```

#### Funkcja czy\_sie\_powtarza

```
bool czy_sie_powtarza(int T[],int poczatek,int koniec)
```

```

{
    int n=koniec-poczatek+1;    //długość podtablicy
    for(int i=0;i<poczatek;i++){    //sprawdzamy, czy podtablica o długości n zaczynająca się w T[i] jest
        taka sama jak tablica o długości n zaczynająca się w T[poczatek]
    }
    int j=0;
    dopóki(T[i+j]==T[poczatek+j]&& j<n)
        j++;
    jeżeli(j==n)
        zwróć true;
    }
    zwróć false;
}

```

#### Funkcja szukaj (główny algorytm)

```
void szukaj(int T[],int n, fstream &plik)
```

```

{
    bool czy_istnieje=0;    //zmienna sprawdzająca czy funkcja znalazła podtablice o sumie 0
    dla(int i=0;dopóki i<n;i++)
    {
        int S=0;    //suma podtablicy
        dla(int j=i;dopóki j<n;j++)
        {
            S+=T[j];
            jeżeli(S==0 and czy_sie_powtarza(T,i,j)==0)    //wypisujemy podtablice, gdy jej suma wynosi 0 i nie
                wypisalimy wcześniej identycznej podtablicy
        }
        jeżeli(czy_istnieje==0)
        {

```

```

wyświetl "Istnieją podtablice, których suma wynosi 0.\nTablice te to:";
zapisz do pliku "Istnieją podtablice, których suma wynosi 0.\nTablice te to:";
czy_istnieje=1;
}
wypisz(T,i,j);
zapisz(T,i,j,plik);
}
}
}
jeżeli(czy_istnieje==0)
{
wyświetl "Brak podtablic, których suma wynosi 0.";
zapisz do pliku "Brak podtablic, których suma wynosi 0.";
}
}

```

### Funkcja test

```
void test(int n, int wmin, int wmax, fstream &plik)
```

```

{
srand(time(NULL));
int T[n];
dla(int i=0;dopóki i<n;i++)
T[i]=rand()%(wmax-wmin+1)+wmin;
wyświetl "   ##### TEST #####\n";
wyświetl "Wygenerowana podtablica to: \n";
wypisz(T,0,n-1);
szukaj(T,n,plik);
wyświetl "   ##### TEST #####\n\n";
}

```

### Główny program

```

int main()
{
fstream plik;
otwórz plik "Dane.txt";

```

```

test(10,-5,5,plik);

int n,tryb;

wyświetl "W jaki sposób chcesz wprowadzić dane?\n1. Z pliku\n2. Z konsoli\nWybierz 1 lub 2: ";

wprowadź tryb;

int *T;

switch(tryb)
{
przypadek 1:
{
fstream dane;

otwórz "Dane_wejsciove.txt";

pobierz z pliku do n;

T=new int[n];

dla(int i=0;dopóki i<n;i++)

dane>>T[i];

zamknij plik;

przerwij;
}

przypadek 2:
{
wyświetl "Ile tablica ma mieć znaków? ";

string tmp;           //zabezpieczenie przed
stringstream ss;      //wprowadzeniem
wprowadź tmp;         //nieprawidłowych
ss<<tmp;              //danych
jeżeli(!(ss>>n))       //mogą być tylko "int"
{
wyświetl "Podano niewłaściwy typ danych";

zamknij plik;

zwróć 1;
}

ss.clear();

```

```

T=new int[n];
dla(int i=0; dopóki i<n; i++)
{
int a; //zmienna pomocnicza
wyświetl "Podaj "<<i+1<<" liczbe: ";
wprowadź tmp;
ss<<tmp;
jeżeli(!(ss>>a))
{
wyświetl "Podano niewlasciwy typ danych"<<endl;
zamknij plik;
zwróć 1;
}
T[i]=a;
ss.clear();
}
przerwij;
}
default:
cout<<"Bład!";
plik.close();
zwróć 404;
}
wyświetl "Zadana tablica: ";
zapisz do pliku "Zadana tablica: ";
wypisz(T,0,n-1);
zapisz(T,0,n-1,plik);
szukaj(T,n,plik);
zamknij plik;
usuń tablicę T;
zwróć 0;
}

```

## 4. Złożoność obliczeniowa

Złożoność obliczeniową tego algorytmu możemy podzielić na dwa scenariusze:

- optymistyczny: w tym przypadku złożoność obliczeniowa wynosi  $n^2$ ;
- pesymistyczny: musimy założyć „najgorszy” scenariusz, w tym przypadku złożoność obliczeniowa wynosi  $n^4$

## 5. Wyniki testów

Przeprowadziłem testy na algorytmie zaczynając od liczby elementów tablicy wynoszącej 100, następnie **1 000**, **10 000**, **25 000** oraz **100 000**, które były wypełnione przez liczby pseudolosowe z zakresu  $[-10,10]$ .

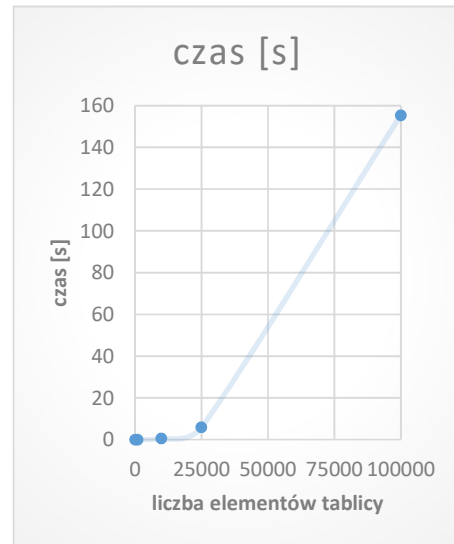
Dla tablicy o **100** elementach czas trwania wyniósł **0s** (prawdopodobnie wynika to z relatywnie małej ilości elementów i dużej mocy obliczeniowej procesora).

Dla tablicy o **1 000** elementów czas wyniósł zaledwie **0,002s**, jest on też oczywiście bardzo mały.

Dla **10 000** elementów czas to **0,5788307s**

Dla **25 000** elementów czas wynosi **5,886s**, widzimy, że jest on już ponad 10 razy większy, niż dla **10 000** elementów.

Dla **100 000** elementów czas oczekiwania jest już bardzo duży, wręcz nieakceptowalny, bo wynosi on aż **155,3978s**.



```
#### TEST ####
Wygenerowana podtablica to:
[ 3, 0, 1, -1, -3, 5, -1, 3, 0, 5 ]
Istnieją podtablice, których suma wynosi 0.
Tablice te to:
[ 3, 0, 1, -1, -3 ]
[ 0 ]
[ 0, 1, -1 ]
[ 1, -1 ]
[ -1, -3, 5, -1 ]
#### TEST ####
```

Rys. 2 Wynik działania programu dla przykładowej tablicy

## 6. Zabezpieczenia

Zabezpieczenia przed wprowadzeniem niewłaściwego typu danych:

```
W jaki sposob chcesz wprowadzic dane?  
1. Z pliku  
2. Z konsoli  
Wybierz 1 lub 2: 0  
Blad!  
Process returned 404 (0x194)    execution time : 6.690 s  
Press any key to continue.
```

Rys. 3 instrukcja switch, wybrałem inną cyfrę niż 1 lub 2

```
W jaki sposob chcesz wprowadzic dane?  
1. Z pliku  
2. Z konsoli  
Wybierz 1 lub 2: 2  
Ile tablica ma miec znakow? x  
Podano niewlasciwy typ danych  
  
Process returned 1 (0x1)    execution time : 5.682 s  
Press any key to continue.
```

Rys. 4 wprowadziłem niewłaściwy typ danych do int n

## 7. Podsumowanie

Podsumowując, program ten dobrze radzi sobie z przeliczaniem na tablicach do 25000 elementów, wraz ze wzrostem rośnie czas oczekiwania na wykonanie się programu. Dla 100 000 elementów czas tej jest wręcz nieakceptowalny i program wymaga poprawek, aby był przydatny do operacji na tak długich tablicach.