SPRAWOZDANIE Z ZADANIA PROJEKTOWEGO

Januszkiewicz Kacper
P03
Inżynieria i analiza danych

Suma najdłuższego podciągu ciągu

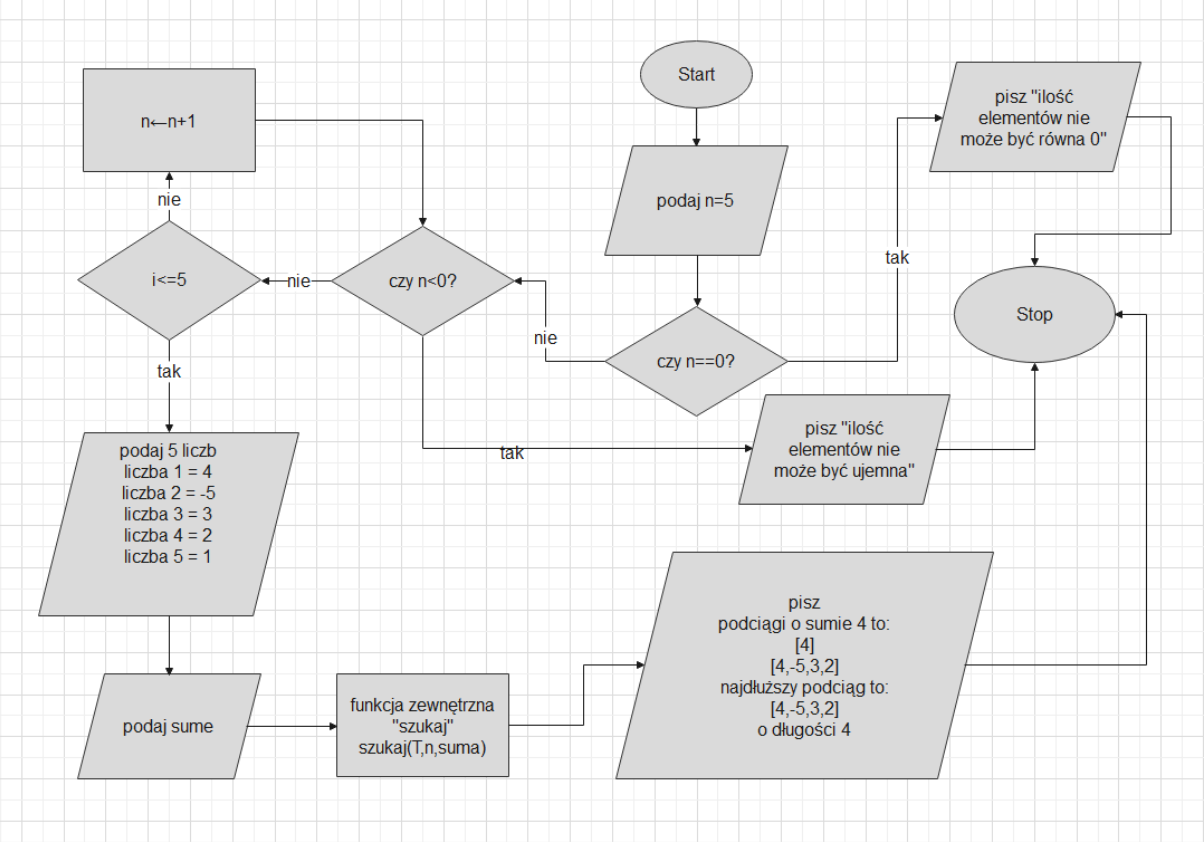
1 Opis problemu

Zadaniem programu jest znalezienie najdłuższego podciągu sposród wszystkich podciągów danego ciągu o zadanej z góry sumie.

2 Schematy blokowe

Ze względu na złożoność programu i na wiele zastosowanych funkcji, schematy blokowe są podzielone na osobne funkcje.

Funkcja główna programu:



3 Pseudokody

T[] - tablica jednowymiarowa
i_start - początkowy indeks tablicy
i_koniec - końcowy indeks tablicy
wprowadź - cin
wyświetl - cout
dla - pętla for

jeżeli - if
funkcja "czy wystąpił podciąg" - czy wystapil podciag - cwp
początek najdłuższego podciągu - pnp
koniec najdłuższego podciągu - knp
w przeciwnym razie - else
szukana suma - szukana_suma
długość ciągu - dlugosc
funkcja testująca program - int testowy
minimalna wartość losowana - minimalna_wartosc
maksymalna wartość losowana - maksymalna_wartosc

3.1 Funkcja „wpisz”

Pseudokod do funkcji „wpisz”

```
int wypisz(int T[],int i_start,int i_koniec)
{
   wyświetl  "[";
   dla (int i=i_start dopóki i<i_koniec;i++)
      wyświetl << T[i]<<", ";
   wyświetl << T[i_koniec] << "]";
}
```

3.2 Funkcja „zapisanie”

Pseudokod do funkcji „zapisane”

```
int zapisanie(int T[],int i_start,int i_koniec,ofstream &plik)
{
   plik << "[";
   dla (int i=i_start dopóki i<i_koniec;i++)
      plik << T[i]<<", ";
   plik << T[i_koniec] << "]"<<endl;
}
```

3.3 Funkcja „cwp”

Pseudokod do funkcji „cwp”

```
int czy wystapil podciag(int T[],int i_start,int i_koniec)
{
   int długość ciągu=i_koniec-i_start+1;
   dla(int i=0 dopóki i<i_start;i++)
      dla(int j=0 dopóki j<długość ciągu;j++){
         jeżeli(T[i+j]różne od T[i_start+j])
```

```
            break;
         jeżeli(j==długość ciągu-1)
            return 1;
      }
   return 0;
}
```

3.4 Funkcja „szukaj”

Pseudokod do funkcji „szukaj”

```
int szukaj(int T[],int n,int szukana suma,ofstream &plik)
{
   int początek najdłuższego podciągu=-1,koniec najdłuższego podciągu=-2;
   wyświetl << "Podciagi o sumie "<< szukana suma<< " to:"<<endl;
   plik << "Podciagi o sumie "<< szukana suma<< " to:"<<endl;
   dla (int i=0 dopóki i<n;i++){
      int S=0;
      dla(int j=i dopóki j<n;j++){
         S+=T[j];
         jeżeli(S==szukana suma&&!czy wystapil podciag(T,i,j))
         {
            wypisz(T,i,j);
            zapisanie(T,i,j,plik);
            jeżeli(j-i>koniec najdłuższego podciągu-początek najdłuższego podciągu){
               początek najdłuższego podciągu=i;
               koniec najdłuższego podciągu=j;
            }
         }
      }
   }
   jeżeli(początek najdłuższego podciągu==-1)
   {
      wyświetl << "Brak takich podciagow"<<endl;
      plik  << "Brak takich podciagow"<<endl;
   }
   w przeciwnym razie{
      wyświetl << "Najdluzszy podciag to:" << endl;
      plik << "Najdluzszy podciag to:" << endl;
      wypisz(T,początek najdłuższego podciągu,koniec najdłuższego podciągu);
      zapisanie(T,początek najdłuższego podciągu,koniec najdłuższego podciągu,plik);
      wyświetl <<"o dlugosci " << (koniec najdłuższego podciągu-początek najdłuższego
podciągu+1)<<endl;
      plik <<"o dlugosci " << (koniec najdłuższego podciągu-początek najdłuższego podciągu+1)<<endl;
   }
}
```

3.5 Funkcja „testowy”

Pseudokod do funkcji „testowy”

```
int funkcja testująca program(int n, int minimalna wartość losowana, int maksymalna wartość losowana,int suma,ofstream &plik)
          {
      srand(time(NULL));
      int T[n];
      dla(int i=0 dopóki i<n;i++)
        T[i]=rand()%(maksymalna wartość losowana-minimalna wartość losowana+1)+minimalna wartość losowana;
      wyświetl<<"Wygenerowany podciag to: "<<endl;
      plik <<"Wygenerowany podciag to: "<<endl;
      wypisz(T,0,n-1);
      zapisanie(T,0,n-1,plik);
      szukaj(T,n,suma,plik);
  }
```

3.6 Funkcja główna programu

Pseudokod do funkcji głównej programu

```
int główna funkcja wykonująca program()
{
   ofstream plik;
   plik.open("plik.txt");
   wyświetl<<"Przyklad z kartki:"<<endl;
   int T1[]={0,6,5,1,-5,5,3,5,3,-2,0};
   szukaj(T1,11,8,plik);
   wyświetl<<endl;
   int suma, n,x=7;
   wyświetl<<"PRZYKLAD"<<endl;
   wyświetl<< "Przykladowy losowy ciag z testera: "<<endl;
   funkcja testująca program(10,-5,5,x,plik);
   wyświetl<<endl<<endl<<endl;
   wyświetl<<"Program obliczy podana suma podanego ciagu"<<endl;
   wyświetl<<"podaj dlugosc tablicy: ";
   wprowadź>>n;
   jeżeli(n==0)
      wyświetl<<"ilosc elementow nie moze byc rowna 0"<<endl;
   else jeżeli (n<0)
      wyświetl<<"ilosc elementow ciagu nie moze byc ujemna"<<endl;
   else
   {
      int T[n];
      wyświetl<<"podaj "<< n << " liczb"<<endl;
      dla(int i=0 dopóki i<n;i++){
```

```
            wyświetl<<"liczba "<<i+1<<": ";
            wprowadź >> T[i];
        }
        wyświetl<<"podaj sume: "<<endl;
        wprowadź>>suma;
        szukaj(T,n,suma,plik);

        wyświetl<<endl;
    }
    plik.close();
    return 0;
}
```

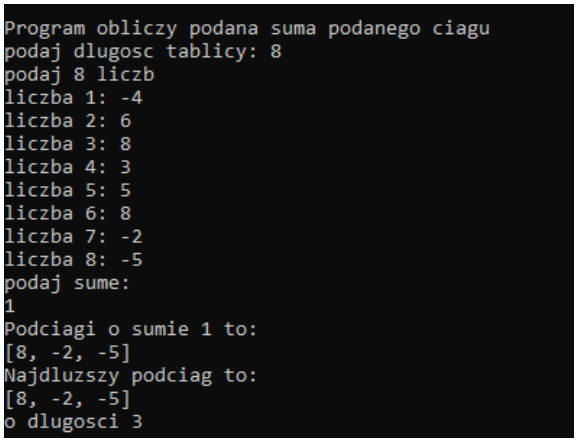
4 Wyniki testów

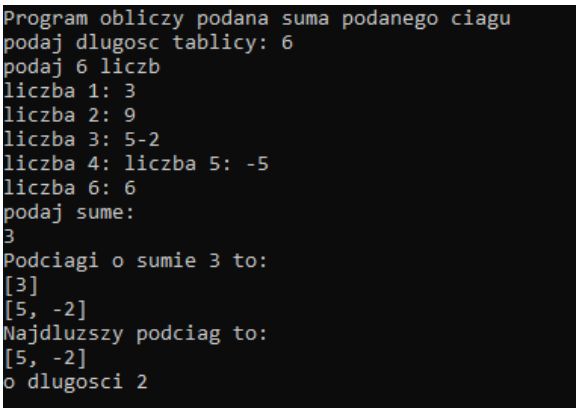
Wyniki przeprowadzanych testów dla większych ilości elementów ciągu tj. 1000, 5000, 10000, 50000 elementów. Wartość elementów ciągu została przypisana losowo przez funkcję „testowe”.

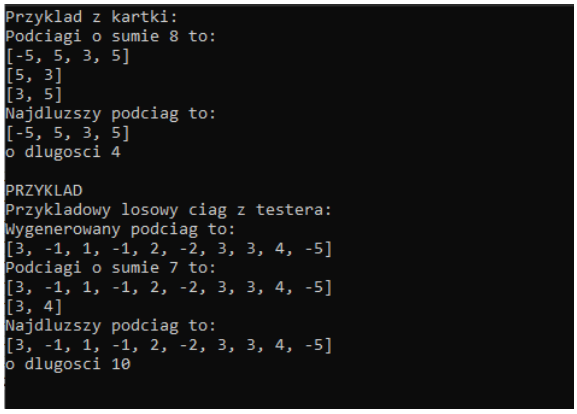
Dla **1000** elementów tablicy czas obliczeniowy wynosi dokładnie 0,0027998 s.
Dla **5000** elementów tablicy czas obliczeniowy wynosi dokładnie 0,3405345 s.
Dla **10000** elementów tablicy czas obliczeniowy wynosi dokładnie 1,04600702 s .
Dla **50000** elementów tablicy czas obliczeniowy wynosi dokładnie 52,02714 s.

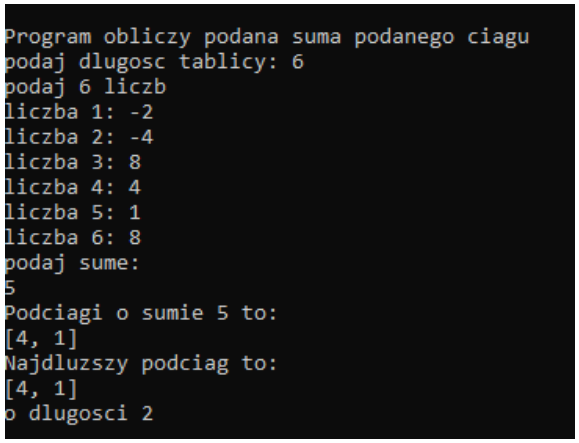
4.1 Testy programu

Testy różnych ciągów przeprowadzone w programie.









5 Złożoność obliczeniowa

Złożoność operacji programu wyliczającego sumę najdłuższego ciągu danej tablicy wynosi n^2 w przypadku optymistycznym, mówiąc natomiast o przypadku pesymistycznym, złożoność wynosi n^4.

6 Podsumowanie

Podsumowując, napisany program sprawdza się dla mniejszej ilości elementów ciągu, gdyż ciąg z 50000 elementami liczył stanowczo za długo. Z drugiej strony, jeśli ma mieć zastosowanie w przeliczaniu ciągów do ~10000 elementowych ciągów, nie powinien sprawiać problemów z długością oczekiwania na wynik.