MIGRA Sp. z o.o.
Skr. poczt. 8
53-350 Wrocław 15

www.migra.pl
biuro@migra.pl

Raport 5

Rozwiązanie zadania 5a

Najlepsza suma: 8

Na podstawie uzyskanego wyniku można podać wartość najlepszej sumy dla ciągu, ponieważ dla celów obliczania najlepszej sumy jest on równoważny poprzedniemu ciągowi. Łatwo bowiem zauważyć, że
2+2+2=6 oraz 3+3+1=7
i po wykonaniu tych dodawań obydwa ciągi są sobie równe.

Rozwiązanie zadania 5b

Najlepsza suma dla *dane5-1.txt*	106
Najlepsza suma dla *dane5-2.txt*	139
Najlepsza suma dla *dane5-3.txt*	1342

Opis algorytmu

Algorytm rozpoczyna się od wczytania liczb z pliku tekstowego o podanej przez użytkownika nazwie do tablicy *ciag* poprzez wywołanie procedury *WczytajDane*. Następnie w funkcji *NajlepszaSuma* znajdowana jest najlepsza suma dla ciągu *ciag* i zwrócona przez tę funkcję wartość jest wypisywana przez program główny. Działanie funkcji *NajlepszaSuma* polega na obliczeniu sumy wszystkich możliwych podciągów ciągu *ciag* i porównywaniu kolejno tak otrzymanych sum z aktualnie największej sumy; jeżeli określona suma jest większa od aktualnie największej sumy, to ta suma staje się aktualnie największą sumą. Wyznaczenie sum wszystkich możliwych podciągów odbywa się w pętlach sterowanych zmiennymi *i* oraz *j*. Wartość zmiennej *i* określa początek podciągu natomiast wartość zmiennej *j* - numer elementu podciągu. Można to zauważyć, że aby obliczyć sumę podciągu o długości *k* i początku i, należy najpierw obliczyć sumę podciągu o długości *k*-1 i początku *i*.

```
program Zadanie5b;
const
      MAX_N=10000;
type
      TCiag=array [1..MAX_N] of integer;
var
      ciag:TCiag;
      n:integer;
      dane:text;

procedure WczytajDane;
var
      nazwaPliku:string;
      dane:text;
begin
      Write('Podaj nazwe pliku: ');
      Readln(nazwaPliku);
      Assign(dane,nazwaPliku);
      Reset(dane);
```

```
        n:=0;
        while not EOF(dane) do
        begin
                Inc(n);
                Readln(dane,ciag[n]);
        end;
        Close(dane);
end;

function NajlepszaSuma:longint;
var
        i,j:integer;
        suma,sumaMax:integer;
begin
        sumaMax:=ciag[1];                       (* Aktualna najwieksza suma *)
        for i:=1 to n do
        begin
                suma:=0;
                for j:=i to n do
                begin
                        Inc(suma,ciag[j]);
                        if suma>sumaMax then
                                sumaMax:=suma;
                end;
        end;
        NajlepszaSuma:=sumaMax;
end;

begin
        WczytajDane;
        Writeln('Najlepsza suma=',NajlepszaSuma);
end.
```

Rozwiązanie zadania 5c

Najpopularniejszy element w *dane5-1.txt*	-22
Najpopularniejszy element w *dane5-2.txt*	-18
Najpopularniejszy element w *dane5-3.txt*	22

Opis algorytmu

Algorytm rozpoczyna się od wczytania liczb z pliku tekstowego podanej przez użytkownika nazwie do tablicy *ciag* poprzez wywołanie procedury *WczytajDane*. Tablica *ciag* jest następnie sortowana algorytmem sortowania szybkiego w procedurze *SortujDane*. Następnie w funkcji *NajpopularniejszyElement* znajdowany jest najpopularniejszy element ciągu *ciag* i zwrócona przez tę funkcję wartość jest wypisywana przez program główny. Działanie funkcji *NajpopularniejszyElement* polega na zliczaniu częstości występowania elementów ciągu *ciag* - ponieważ elementy te zostały uprzednio posortowane, wystarczy więc zwiększać licznik wystąpienia elementu o 1 jeżeli wartość elementu nie zmieniła się lub ustawić wartość tego licznika na 1 jeżeli wartość elementu zmieniła się. W tym pierwszym przypadku następuje dodatkowo sprawdzenie, czy nowa częstość jest większa od aktualnie maksymalnej częstości i jeśli tak, to ta częstość staje się aktualnie maksymalną częstością oraz wartość danej zostaje zapisana w zmiennej *maxWart*.

Oszacowanie czasu działania

Liczba operacji *F(n)* niezbędna dla wykonania algorytmu dla zbioru o rozmiarze n wynosi $F(n) \approx Fs(n)+n$, gdzie *Fs(n)* – liczba operacji dla algorytmu sortowania szybkiego $Fs(n)=n*\log_2 n$.

```
program Zadanie5c;
const
      MAX_N=10000;
type
      TCiag=array [1..MAX_N] of integer;
var
      ciag:TCiag;
      n:integer;
      dane:text;

procedure WczytajDane;
var
      nazwaPliku:string;
      dane:text;
begin
      Write('Podaj nazwe pliku: ');
      Readln(nazwaPliku);
      Assign(dane,nazwaPliku);
      Reset(dane);
      n:=0;
      while not EOF(dane) do
      begin
            Inc(n);
            Readln(dane,ciag[n]);
      end;
      Close(dane);
end;

(* Sortowanie szybkie, por. Wirth *)
procedure SortujDane(l,p:integer);
var
      i,j:integer;
      x,w:integer;
begin
      i:=l; j:=p;
      x:=ciag[(l+p) div 2];
      repeat
            while ciag[i]<x do Inc(i);
            while x<ciag[j] do Dec(j);
            if i<=j then
            begin
                  w:=ciag[i]; ciag[i]:=ciag[j]; ciag[j]:=w;
                  Inc(i);     Dec(j);
            end;
      until i>j;
      if l<j then SortujDane(l,j);
      if i<p then SortujDane(i,p);
end;

function NajpopularniejszyElement:integer;
var
      maxCzest,maxWart:integer;
```

```
        czest:integer;
        i:integer;
begin
        (* Ustawienie wartosci poczatkowych *)
        maxCzest:=1;
        maxWart:=ciag[1];
        czest:=1;
        for i:=2 to n do
                if ciag[i]<>ciag[i-1] then    (* Wartosc zmienila sie *)
                begin
                        if czest>maxCzest then
                        begin
                                maxCzest:=czest;
                                maxWart:=ciag[i-1];
                        end;
                        czest:=1;
                end
                else
                        Inc(czest);
        NajpopularniejszyElement:=maxWart;
end;

begin
        WczytajDane;
        SortujDane(1,n);
        Writeln('Najpopularniejszy element=',NajpopularniejszyElement);
end.
```