

Algorytmy i struktury danych

Projekt

P01 Kamil Bieniek

Opis problemu

Dla zadanego ciągu zer i jedynek, znajdź wszystkie dłuższe niż dwuelementowe podciągi, w których zera poprzedza ta sama liczba jedynek.

Przykład:

Wejście [0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0]

Wyjście [1, 1, 1, 0, 0, 0] , [1, 1, 0, 0]

Wejście [0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1]

Wyjście Brak elementów spełniających zadane kryteria.

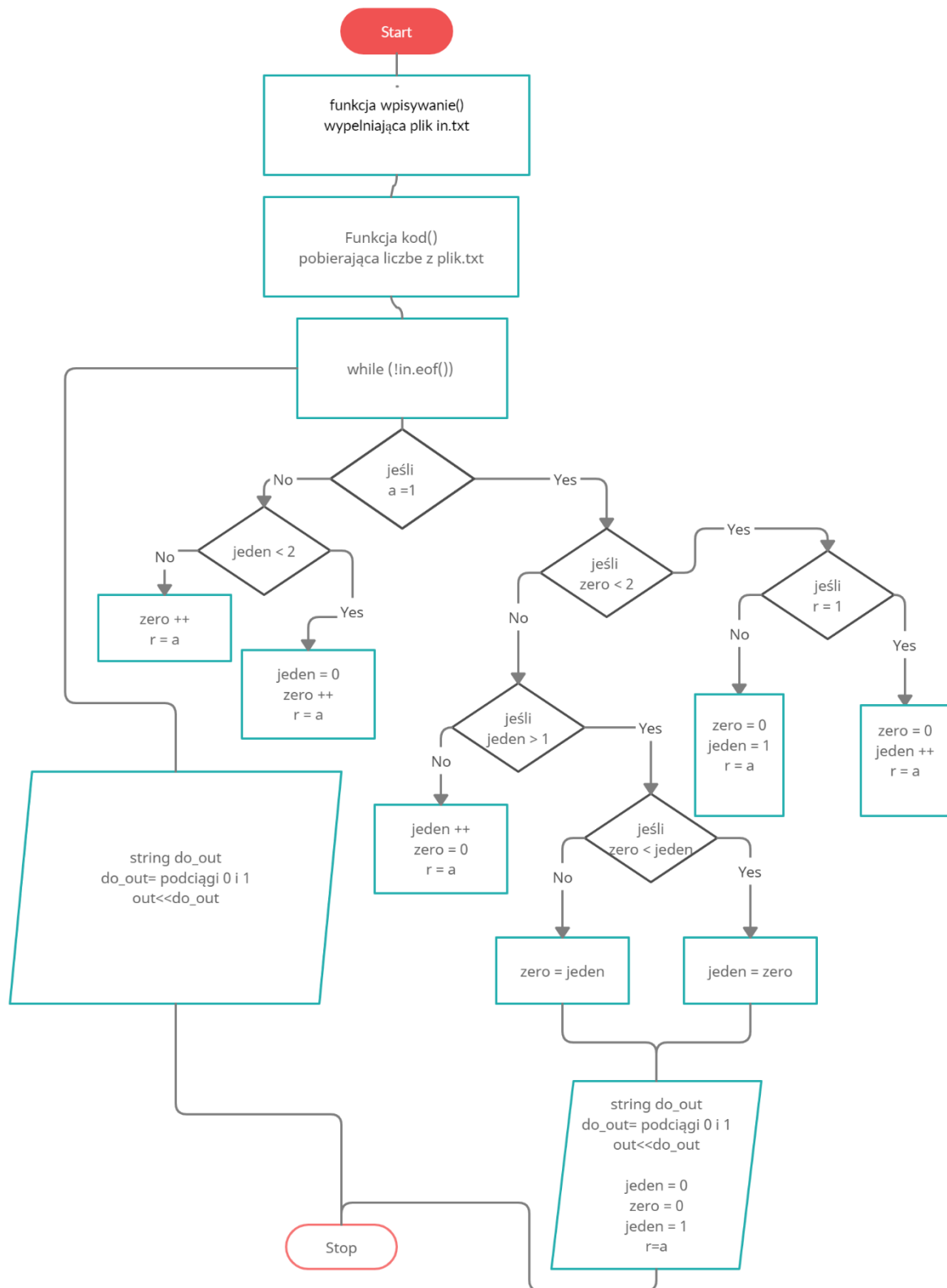
Funkcje użyte w programie

1. wpisywanie();
 - Wpisuje ciąg losowych liczb z zakresu 0 – 1 do pliku in.txt
2. kod(z);
 - wyciąga z pliku in.txt podciągi spełniające kryteria i wpisuje je do pliku out.txt

Opis działania programu

Program ma za zadanie wpisanie do pliku in.txt losowych wygenerowanych liczb z zakresu 0 – 1 w ilości podanej przez użytkownika. Następnie funkcja kod(z) wyciąga z pliku podciągi zer i jedynek w których liczba zer jest taka sama jak liczba jedynek poprzedzająca je a następnie wpisuje je do pliku w nawiasach kwadratowych. W programie jest też zawarty kod zliczający nam czas wykonywanie całego kodu w mikro sekundach.

Schemat Blokowy



Pseudokod

```
wpisywanie();
    wpisanie do pliku in.txt losowych liczb z przedziału 0-1
kod(z);
    in= dane z pliku in.txt
    dopóki (!in.eof())
        in>>a;
        jeżeli(a==1)
            {jeżeli(zero<2)
                {jeżeli(r==1)
                    {zero = 0;jeden ++;r=a;}
                w przeciwnym wypadku
                    {zero=0;jeden=1;r=a;}
            }
        w przeciwnym wypadku
            {jeżeli(jeden>1)
                {jeżeli(zero>jeden
                    {jeden=zero;}
                w przeciwnym wypadku
                    {zero=jeden;}

                dla(i=0;i<jeden;i++)
                    {output=output+"1,"}
                dla(i=0;i<zero;i++)
                    {output=output+"0,"}output=output+"]"
                out << do_out ;jeden=0;zero=0;jeden++;r=a;}
                w przeciwnym wypadku
                    {jeden++;zero=0;r=a;}
            }
        w przeciwnym wypadku{
            jeśli(jeden<2)
                {jeden=0;
                    zero++;
                    r=a;}
            w przeciwnym wypadku
                {zero++;
                    r=a;}
        }
    jeżeli(zero-1>1 i jeden>1)
        {jeśli{zero>jeden}
            {jeden=zero;}
        w przeciwnym wypadku
            {zero=jeden;}
    dla(i=0;i<jeden;i++)
        {output=output+"1,";}
    dla(i=0;i<zero;i++)
        {output=output+"0,";} output=output+"]";
    out<<output;
```

Czas działanie algorytmu w mikrosekundach

| | |
|---------------------|---------------------|
| Dla 10 elementów | 707530 mikrosekund |
| Dla 100 elementów | 855717 mikrosekund |
| Dla 1000 elementów | 1278575 mikrosekund |
| Dla 10000 elementów | 2257012 mikrosekund |

