

Akademia Nauk Stosowanych - Teoretyczne i technologiczne podstawy multimediiów - laboratorium			
Temat: Algorytm LZ77.			Symbol: TiTPM
Nazwisko i imię: Fyda Kamil		Ocena sprawozdania	Zaliczenie:
Data wykonania ćwiczenia: 22.11.2022r.		Grupa: L1	

1. Opis teoretyczny:

LZ77 (Lempel-Ziv 77) – metoda strumieniowej słownikowej kompresji danych. Metoda wykorzystuje fakt iż w danych przeznaczonych do zakodowania powtarzają się pewne ciągi bajtów/znaków (zależy jaką formą danych wejściowych się posługujemy). Algorytm LZ77 jest wolny od wszelkich patentów, co w dużej mierze przyczyniło się do jego popularności i szerokiego rozpowszechnienia. Doczekał się wielu ulepszeń i modyfikacji, dających lepsze współczynniki kompresji albo dużą szybkość działania. Na LZ77 opiera się m.in. algorytm deflate, używany jest również w formatach ZIP, gzip, ARJ, RAR, PKZIP, a także PNG. Algorytm został opracowany w 1977 przez Abrahama Lempela i Jacoba Ziv i opisany w artykule A opublikowanym w IEEE. Rok później autorzy opublikowali ulepszoną wersję metody, znaną pod nazwą LZ78. Organizacja IEEE uznała algorytm Lempel-Ziv za kamień milowy w rozwoju elektroniki i informatyki.

2. Algorytm:

1. Przyjmijmy rozmiar bufora na n oraz rozmiar słownika na m .
2. Wypełnijmy bufor pierwszymi n znakami.
3. W słowniku szukamy najdłuższego możliwego prefikсового ciągu znaków, który znajduje się w słowniku.
4. Jeżeli nie znajdziemy dopasowania to zapisujemy wartość (i, j, k) gdzie i oraz j to 0 a k to pierwszy symbol w buforze. Następnie przesuwamy całość o 1 w lewo.
5. Jeżeli znajdziemy dopasowanie to zapisujemy wartość (i, j, k) gdzie i to indeks znalezionego ciągu, j to rozmiar znalezionego podciągu, k to pierwszy znak za znalezionym ciągiem. Następnie przesuwamy się o $j+1$ w lewo.
6. Wracamy do punktu 3 jeżeli w buforze znajdują się jakieś dane.

3. Kod programu:

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main()
{
    string tablica[5];
    string wyjscie[10][3];
    cout << "LZ77 - algorytm kodowania\n";
    string text = "ababcbababaaaaa";
    int n = 4;
    int m = 4;
    string slownik = "____";
    string bufor = text.substr(0, 4);
    int przesuniecie = 0;
    int calosc = 0;

    int zerowy = 0;
    int dlugosc = 0;
    int index = 0;
    int zzz = 0;
    while (calosc <= text.length())
    {
        cout << " Slownik: " << slownik << " Bufor " << bufor << endl;
        int przesuniecie = 0;

        int flaga = 0;

        for (int i = n; i > 0; i = i - 1)
        {
            string bufor2 = bufor.substr(0, i);
            int found = slownik.find(bufor2);
            if (found != string::npos)
            {
                index = found;

                dlugosc = bufor2.length();
                flaga = 1;
                break;
            }
        }

        if (flaga == 1)
        {
            wyjscie[zzz][0] = to_string(index);
            wyjscie[zzz][1] = to_string(dlugosc);
            wyjscie[zzz][2] = bufor[dlugosc];

            string stary_slownik = slownik + text.substr(calosc, text.length());

            przesuniecie = przesuniecie + dlugosc;
            calosc = calosc + przesuniecie + 1;
            for (int y = 0; y < n; y++)
                slownik[y] = stary_slownik[y + przesuniecie + 1];
            if (calosc < text.length())
                bufor = text.substr(calosc, 4);
        }
        if (flaga == 0)
        {
            wyjscie[zzz][0] = "0";
            wyjscie[zzz][1] = "0";
            wyjscie[zzz][2] = bufor[0];
            string stary_slownik = slownik + text.substr(calosc, text.length());
            przesuniecie++;
            calosc = calosc + przesuniecie;
            for (int y = 0; y < n; y++)
                slownik[y] = stary_slownik[y + 1];
            if (calosc < text.length())
                bufor = text.substr(calosc, 4);
        }
        zzz++;
    }

    for (int i = 0; i < 10; i++)
    {
        cout << wyjscie[i][0] << " " << wyjscie[i][1] << " " << wyjscie[i][2] << endl;
    }
}
```

```


cout << "LZ77 - algorytm dekodowania\n";
string tekst = "";
string wejscie[10][3];
ifstream plik("dane.txt");
string linia;
int aaa = 0;
int i = 0, j = 0;
string a, b, c;
string k = "";
slownik = "";
while (getline(plik, linia)) \
{
    a = linia[0];
    b = linia[2];
    k = linia[4];
    i = stoi(a);
    j = stoi(b);
    slownik = linia.substr(6, 4);

    string wycinamy = slownik.substr(i, j);
    tekst = tekst + wycinamy + k;
    cout << i << " " << j << " " << k << " " << slownik << endl;

    aaa++;
}
cout << "Tekst: " << tekst << endl;
plik.close();
}

```

4. Wynik działania programu:


 Konsola debugowania programu Microsoft Visual Studio

```

LZ77 - algorytm kodowania
Slownik: ____ Bufor abab
Slownik: __a Bufor babc
Slownik: __ab Bufor abcb
Slownik: babc Bufor baba
Slownik: baba Bufor baaa
Slownik: abaa Bufor aaaa
Slownik: aaaa Bufor a
0 0 a
0 0 b
2 2 c
0 3 a
0 2 a
2 2 a
0 1

LZ77 - algorytm dekodowania
0 0 a
0 0 b a
2 2 c ab
0 3 a babc
0 2 a baba
2 2 a abaa
0 1 aaaa
Tekst: ababcbababaaaaaa

```

 dane — Notatnik

Plik Edycja Format Widok Pomoc

```

0 0 a
0 0 b a
2 2 c ab
0 3 a babc
0 2 a baba
2 2 a abaa
0 1 aaaa

```