Lab - 7
CT-216

LEMPEL ZIV ENCODING/DECODING

Rakshit Pandhi-202201426

```
% Rakshit Pandhi - 202201426

% LEMPEL ZIV ENCODING AND DECODING
```

```
load('discreteSources.mat')
s=char('0'+S1)
```

```
s =
'113114211131121321111511111213152541121522113221112113421213121111133212131232122111121121451113126231213121112431
```

```
% Using map library
% Basically map will work as a dictionary (motivation of map from CPP)
mp= containers.Map;

% Making a string/substring as current which will basically parse the
% original string

current='';
addr=1;
arr={};

% Initialize the first substring/string to 0
mp(current)=0;

for i=1:numel(s)
    current=[current,s(i)];

    if ~isKey(mp,current) || mp(current)==0
        arr{end+1}=current;
        mp(current)=addr;
        addr=addr+1;
        current='';
    end
end
```

```
% Encoding part

addr=addr-1;

% No. of bits required for storing max. value
size=ceil(log2(addr));

% Encoded string
encoded='';

for i=1:numel(arr)
    temp=arr{i};
    temp(end)=[];
```

```
    dc=dec2bin(mp(temp),size);
    encoded=[encoded,dc,arr{i}(end)];
end

disp('Encoded string is');
```

```
Encoded string is
```

```
disp(encoded);
```

```
0000000000000000010000000000000013000000000000001100000000000000040000000000000002000000000000001110000000
```

```
% Decoding part

% Decoded string
decoded='';

for i=1:(size+1):numel(encoded)
    temp=0;
    for j=1:size
        if encoded(-1+i+j)=='1'
            temp=temp+(2^(size-j)); % this bascially gives binary val its decimal
value related to its postion (bin2dec bascially)
        end
    end
    if temp~=0
        decoded=[decoded,arr{temp}];
    end
    decoded=[decoded,encoded(i+size)];
end

% Append any left substring from the encoding process
% This is necessary in case of no new phrases for e.g. 11111

decoded=[decoded,current];
disp('Final decoded string is');
```

```
Final decoded string is
```

```
disp(decoded);
```

```
113114211131121321111511111213152541121522113221112113421213121111133212131232122111121121451113126231213121112431313
```

Also attaching C++ Code which I wrote first to test with and then changed it to a MATLAB Code

```cpp
#include<bits/stdc++.h>
using namespace std;
#define ll long long
string ConvertDecimalToBinary(int number)
{
    string binaryString;
    if(number == 0){
        return "0";
    }
    while(number){
      if(number & 1) // 1
        binaryString += '1';
      else // 0
        binaryString += '0';
      number >>= 1; // Right Shift by 1
    }
    reverse(binaryString.begin(), binaryString.end());
    return binaryString;
}
int main(){
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    string inputString = "1111010101";
    map<string,int> mapping;
    string current = "";
    int value = 1;
    vector<string> substrings;
    mapping[current] = 0;
    for(int i = 0; i < inputString.size(); i++){
        current += inputString[i];
        if(mapping[current] == 0){
            substrings.push_back(current);
            mapping[current] = value;
            value++;
            current = "";
        }
    }
```

```
    cout << value << endl;
    value--;
    int size = ceil(log2(value));
    string encoded = "";
    for(int i = 0; i < substrings.size(); i++){
        string temp = substrings[i];
        temp.pop_back();

        string binaryRep = ConvertDecimalToBinary(mapping[temp]);
        for(int k = 0; k <= size - binaryRep.size(); k++){
            binaryRep = '0' + binaryRep;
        }
        encoded += binaryRep;
        encoded += substrings[i][substrings[i].size() - 1];
    }
    cout << encoded << endl;
    string decoded = "";
    for(int i = 0; i < encoded.size(); i += (size + 1)){
        int temp = 0;
        for(int j = 0; j < size; j++){
            if(encoded[i + j] == '1'){
                temp += (1 << (2 - j));
            }
        }
        cout << temp << endl;
        if(temp != 0){
            decoded += substrings[temp - 1];
        }
        decoded += encoded[i + 3];
    }
    decoded += current;
    cout << decoded << endl;
    return 0;
}
```