

CSE-3024 Web Mining

Lab Assignment 5

Alokam Nikhitha

19BCE2555

Question

Experiment 5

Write a python program to perform the following encoding and decoding for the EVEN numbers between 1-20

- 1) Unary
- 2) Elias Gamma
- 3) Elias Delta
- 4) Golomb (b=10)

Problem statement:

To perform the following encoding and decoding for the EVEN numbers between 1-20

Procedure:

- Firstly, we will import the necessary numpy library to use mathematical functions like logarithm in our code.
- Next, We will create 2 functions, one to convert integer to binary and the other for converting binary to integer.
- Next, We will write respective functions for each and every method given.
- The functions that are corresponding to Unary Encoding, Unary Decoding, Elias Gamma Encoding, Elias Gamma Decoding, Elias Delta Encoding, Elias Delta Decoding, Golomb Encoding and Golomb Decoding.
- In main program, we will run a loop from numbers 2 to 21 with a jump of 2 to in order to get even numbers in the range from 1-20.
- We will finally perform the above functions to each of the iterators in the above loop

Code:

```
In [1]: #19BCE2555
#Importing Library
import numpy as np
```

```
In [2]: #Converting Integer to Binary
def intToBin(var):
    return bin(var).split("0b")[1]
```

```
In [3]: #Converting Binary to Integer
def binToInt(var):
    return int(var, 2)
```

```
In [4]: #Unary Encoding
def unaryEncoding(var):
    unary = ""
    for i in range(var-1):
        unary='0'+unary
    unary=unary+'1'
    return unary
```

```
In [5]: #Unary Decoding
def unaryDecoding(var):
    counter=0
    while(var[0]=='0'):
        var=var[1:]
        counter=counter+1
    return counter+1
```

```
In [6]: #Elias Gamma Encoding
def eliasGammaEncoding(var):
    var = intToBin(var)
    n=len(var)-1
    for i in range(n):
        var = '0'+var
    return var
```

```
In [7]: #Elias Gamma Decoding
def eliasGammaDecoding(var):
    counter=0
    while(var[0]=='0'):
        var=var[1:]
        counter=counter+1
    var=var[0:counter+1:1]
    return binToInt(var)
```

```
In [8]: #Elias Delta Encoding
def eliasDeltaEncoding(var):
    selector = eliasGammaEncoding(1+int(np.log2(var)))
    var = intToBin(var)
    offset=""
    for i in range(1, len(var)):
        offset=offset+var[i]
    return (selector+offset)
```

```
In [9]: #Elias Delta Decoding
def eliasDeltaDecoding(var):
    Nbits=eliasGammaDecoding(var)-1
    ans=""
    for i in range(Nbits):
        ans=var[-(i+1)]+ans
    return binToInt('1'+ans)
```

```
In [10]: #Golomb Encoding
def golombEncoding(var, b):
    quotient=unaryEncoding(int(var/b) +1)
    remainder=var%b
    i=int(np.log2(b))
    d= (2**(i+1))-b
    if (remainder<d):
        r = intToBin(remainder)
        while len(r)<i:
            r='0'+r
    else:
        r=intToBin(remainder+d)
        while len(r)<i+1:
            r='0'+r
    return quotient+r
```

```
In [11]: #Golomb Decoding
def golombDecoding(var, b):
    quotient=unaryDecoding(var)-1
    i=int(np.log2(b))
    d=(2**(i+1))-b
    counter=0
    while (var[0]!='0'):
        var=var[1:]
        counter=counter+1
    var=var[1:]
    remainder=var[0:i]
    remainder=binToInt(remainder)
    if (remainder>=d):
        remainder=intToBin(remainder)
        remainder=var[0:i+1]
        remainder=binToInt(remainder)-d
    ans=quotient*b+remainder
    return ans
```

```

In [12]: for i in range(2,21,2):
          print("\n\nNumber=",i)
          UE = unaryEncoding(i)
          print("\tUnaryEncoding: ", UE)
          EGE=eliasGammaEncoding(i)
          print("\tElias Gamma Encoding: ",EGE)
          EDE=eliasDeltaEncoding(i)
          print("\tElias Delta Encoding: ",EDE)
          GE=golombEncoding(i,10)
          print("\tGoloumb Encoding: ",GE)
          print("\tUnary Decoding:", unaryDecoding(UE))
          print("\tElias Gamma Decoding:", eliasGammaDecoding(EGE))
          print("\tElias Delta Decoding:", eliasDeltaDecoding(EDE))
          print("\tGolomb Decoding:", golombDecoding(GE,10))

```

Code Snippets and Outputs:

```

In [1]: #19BCE2555
        #Importing Library
        import numpy as np

```

Here we are importing the libraries that are required.

```

In [2]: #Converting Integer to Binary
        def intToBin(var):
            return bin(var).split("0b")[1]

```

```

In [3]: #Converting Binary to Integer
        def binToInt(var):
            return int(var, 2)

```

```

In [4]: #Unary Encoding
        def unaryEncoding(var):
            unary = ""
            for i in range(var-1):
                unary='0'+unary
            unary=unary+'1'
            return unary

```

```
In [5]: #Unary Decoding
def unaryDecoding(var):
    counter=0
    while(var[0]=='0'):
        var=var[1:]
        counter=counter+1
    return counter+1
```

```
In [6]: #Elias Gamma Encoding
def eliasGammaEncoding(var):
    var = intToBin(var)
    n=len(var)-1
    for i in range(n):
        var = '0'+var
    return var
```

```
In [7]: #Elias Gamma Decoding
def eliasGammaDecoding(var):
    counter=0
    while(var[0]=='0'):
        var=var[1:]
        counter=counter+1
    var=var[0:counter+1:1]
    return binToInt(var)
```

```
In [8]: #Elias Delta Encoding
def eliasDeltaEncoding(var):
    selector = eliasGammaEncoding(1+int(np.log2(var)))
    var = intToBin(var)
    offset=""
    for i in range(1, len(var)):
        offset=offset+var[i]
    return (selector+offset)
```

```
In [9]: #Elias Delta Decoding
def eliasDeltaDecoding(var):
    Nbits=eliasGammaDecoding(var)-1
    ans=""
    for i in range(Nbits):
        ans=var[-(i+1)]+ans
    return binToInt('1'+ans)
```

```

In [10]: #Golomb Encoding
def golombEncoding(var, b):
    quotientunary=unaryEncoding(int(var/b) +1)
    remainder=var%b
    i=int(np.log2(b))
    d= (2**(i+1))-b
    if (remainder<d):
        r = intToBin(remainder)
        while len(r)<i:
            r='0'+r
    else:
        r=intToBin(remainder+d)
        while len(r)<i+1:
            r='0'+r
    return quotientunary+r

```

```

In [11]: #Golomb Decoding
def golombDecoding(var, b):
    quotient=unaryDecoding(var)-1
    i=int(np.log2(b))
    d=(2**(i+1))-b
    counter=0
    while (var[0]=='0'):
        var=var[1:]
        counter=counter+1
    var=var[1:]
    remainder=var[0:i]
    remainder=binToInt(remainder)
    if (remainder>=d):
        remainder=intToBin(remainder)
        remainder=var[0:i+1]
        remainder=binToInt(remainder)-d
    ans=quotient*b+remainder
    return ans

```

Here, we had defined all the ten functions that are described in procedure.

```
In [12]: for i in range(2,21,2):
          print("\n\nNumber=",i)
          UE = unaryEncoding(i)
          print("\tUnary Encoding: ", UE)
          EGE=eliasGammaEncoding(i)
          print("\tElias Gamma Encoding: ",EGE)
          EDE=eliasDeltaEncoding(i)
          print("\tElias Delta Encoding: ",EDE)
          GE=golombEncoding(i,10)
          print("\tGolomb Encoding: ",GE)
          print("\tUnary Decoding:", unaryDecoding(UE))
          print("\tElias Gamma Decoding:", eliasGammaDecoding(EGE))
          print("\tElias Delta Decoding:", eliasDeltaDecoding(EDE))
          print("\tGolomb Decoding:", golombDecoding(GE,10))
```

```
Number= 2
    UnaryEncoding: 01
    Elias Gamma Encoding: 010
    Elias Delta Encoding: 0100
    Golomb Encoding: 1010
    Unary Decoding: 2
    Elias Gamma Decoding: 2
    Elias Delta Decoding: 2
    Golomb Decoding: 2

Number= 4
    UnaryEncoding: 0001
    Elias Gamma Encoding: 00100
    Elias Delta Encoding: 01100
    Golomb Encoding: 1100
    Unary Decoding: 4
    Elias Gamma Decoding: 4
    Elias Delta Decoding: 4
    Golomb Decoding: 4
```

Here we are running a loop in order to iterate the even numbers in range 1-20 and then use the above functions to get our results.

Results and Output

```
Number= 2
    UnaryEncoding: 01
    Elias Gamma Encoding: 010
    Elias Delta Encoding: 0100
    Golomb Encoding: 1010
    Unary Decoding: 2
    Elias Gamma Decoding: 2
    Elias Delta Decoding: 2
    Golomb Decoding: 2
```


Number= 4

UnaryEncoding: 0001
Elias Gamma Encoding: 00100
Elias Delta Encoding: 01100
Golomb Encoding: 1100
Unary Decoding: 4
Elias Gamma Decoding: 4
Elias Delta Decoding: 4
Golomb Decoding: 4

Number= 6

UnaryEncoding: 000001
Elias Gamma Encoding: 00110
Elias Delta Encoding: 01110
Golomb Encoding: 11100
Unary Decoding: 6
Elias Gamma Decoding: 6
Elias Delta Decoding: 6
Golomb Decoding: 6

Number= 8

UnaryEncoding: 00000001
Elias Gamma Encoding: 0001000
Elias Delta Encoding: 00100000
Golomb Encoding: 11110
Unary Decoding: 8
Elias Gamma Decoding: 8
Elias Delta Decoding: 8
Golomb Decoding: 8

Number= 10

UnaryEncoding: 0000000001
Elias Gamma Encoding: 0001010
Elias Delta Encoding: 00100010
Golomb Encoding: 01000
Unary Decoding: 10
Elias Gamma Decoding: 10
Elias Delta Decoding: 10
Golomb Decoding: 10

Number= 12

UnaryEncoding: 000000000001
Elias Gamma Encoding: 0001100
Elias Delta Encoding: 00100100
Golomb Encoding: 01010
Unary Decoding: 12
Elias Gamma Decoding: 12
Elias Delta Decoding: 12
Golomb Decoding: 12

Number= 14

UnaryEncoding: 00000000000001
Elias Gamma Encoding: 0001110
Elias Delta Encoding: 00100110
Golomb Encoding: 01100
Unary Decoding: 14
Elias Gamma Decoding: 14
Elias Delta Decoding: 14
Golomb Decoding: 14

Number= 16

UnaryEncoding: 0000000000000001
Elias Gamma Encoding: 000010000
Elias Delta Encoding: 001010000
Golomb Encoding: 011100
Unary Decoding: 16
Elias Gamma Decoding: 16
Elias Delta Decoding: 16
Golomb Decoding: 16

Number= 18

UnaryEncoding: 000000000000000001
Elias Gamma Encoding: 000010010
Elias Delta Encoding: 001010010
Golomb Encoding: 011110
Unary Decoding: 18
Elias Gamma Decoding: 18
Elias Delta Decoding: 18
Golomb Decoding: 18

Number= 20

UnaryEncoding: 00000000000000000001

Elias Gamma Encoding: 000010100

Elias Delta Encoding: 001010100

Golomb Encoding: 001000

Unary Decoding: 20

Elias Gamma Decoding: 20

Elias Delta Decoding: 20

Golomb Decoding: 20