23/01/23

Modular Arithmetic

$$a \bmod n = a - n * \left\lfloor \frac{a}{n} \right\rfloor$$

$x + y = 0$, x & y are additive inverse of each other

$(x+y) \bmod n = 0$

$x \times y = 1$, x & y are multiplicative inverse of each other.

$(x \times y) \bmod n = 1$

If $\gcd(m, a) = 1$, then only M.I can be calculated.

Euclidean Algorithm of gcd

```
r1 ← a ; r2 ← b
while (r2 > 0) {
    q ← r1/r2;
    r ← r1 - q × r2;
    r1 ← r2, r2 ← r
}
```

$r_1 = 2740$

$r_2 = 1760$

$q = r_1 / r_2$

$r = r_1 - q \times r_2$

Extended Euclidean Algo.

$S \times a + t \times b$

Given $a = 161$, $b = 28$

find $\gcd(a, b)$

$S = -1$

$t = 6$

Ans = $\gcd(161, 28) = 7$

A linear Diophantine equation of 2 variables is $ax + by = c$. It either has no solution or ∞ no. of solutions.

Particular solution:

1) Reduce the eqn. to $a_1 x + b_1 y = c_1$ by dividing b.s by d.

2) Solve for s and t in the relation $a_1 s + b_1 t = 1$

3) The particular solution can be found.

If $d \nmid c$ (d doesn't divide c), eqn. has no solution

If $d \mid c$, eqn. has infinite solutions.

Particular solution:

$$x_0 = (c/d)s$$
$$y_0 = (c/d)t$$

General solution:

$$x = x_0 + k(b/d)$$
$$y = y_0 - k(a/d) \quad \text{where } k \text{ is an integer}$$

Qn) Find the particular and general solutions to the equation $21x + 14y = 35$

$$s = 1, \quad t = -1$$
$$d = \gcd(21, 14) = 7$$

$$x_0 = \left(\frac{35}{7}\right) 1 = 5 \times 1 = 5$$

$$y_0 = \left(\frac{35}{7}\right) \times (-1) = 5 \times (-1) = -5$$

General solution: $x = 5 + k\left[\frac{14}{7}\right] = 5 + k[2]$
$= 5 + 2k.$

$$y = -5 \mp k\left[\frac{21}{7}\right] = -5 \mp 3k$$

If $k = 1$, Solution $= (7, -8)$

$k = 2$, Solution $= (9, -11)$

Congruence

To show 2 integers are congruent, we use $\equiv$ operator.

$2 \equiv 12 \pmod{10}$
$3 \equiv 8 \pmod{5}$
$13 \equiv 23 \pmod{10}$
$8 \equiv 13 \pmod{5}$

$a \equiv b \bmod n$ iff $|a-b|$ is divisible by n.

$-6 \equiv 1 \bmod 7$

$\frac{-1}{2} \bmod 23 =$

$\frac{-1}{2} - 23 * \left\lfloor \frac{1/2}{23} \right\rfloor$

$\frac{-1}{2} - \frac{1}{2}$

$= -1$

23×1
$46\ 2$

$2^{-1} \bmod 23 = 12$

$-12 \bmod 23 = 11$

$\frac{-1}{4} \bmod 23 = 4^{-1} \bmod 23 = 6$
$-6 \bmod 23 = 17$

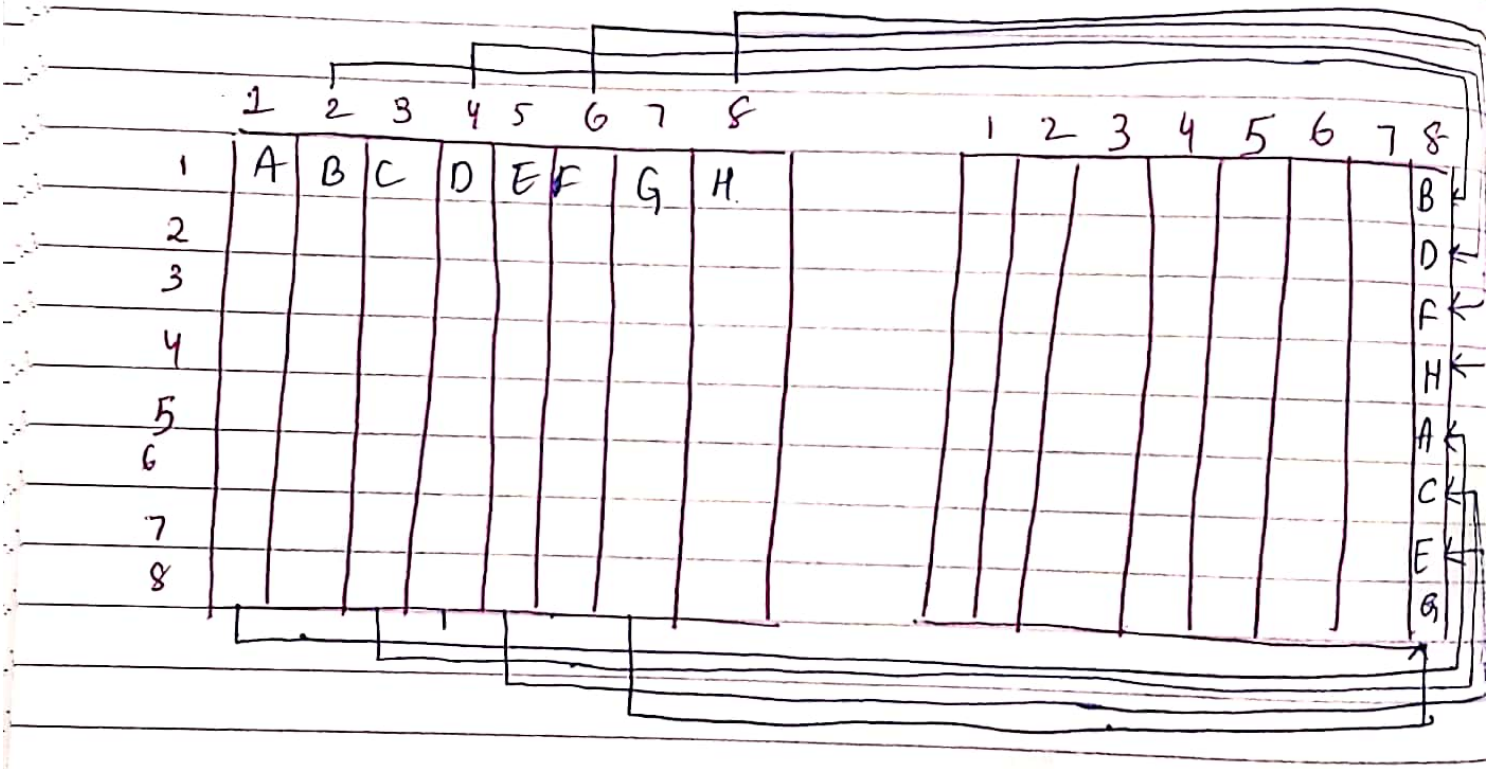
8/2/23

The bit in the first octet of input get spread into the 8 bits of each of the octets.

The bits in the 2nd octets of the input get spread into the 7th bit of the octet.

In general, the bits of the ith octet get spread into the ~~(bits)~~ (8-i)+1 bits of all the octets.

The pattern of the spreading of the 8th bit in octet 'i' of the input among the output octet is that even no. bits go into octet 2 to 4. and the odd no. bits go into octet 5 to 8.



Example 6.1

Example 6.2

- Rounds in DES
→ DES Function
→ Expansion P-box

Confusion → S-box
Diffusion → Transfusion

4 5 6 7 8 9

→ Avalanche effect.

NetSec – AES page 215-232.

→ General design of AES encryption cipher
→ Data units in AES.
Example 7.1.

H E B A
08 07 04 01 00

07
04
01
00

H → 8
8
1000

18
10010
1 2

Structure of each round.
SubBytes → Substitution Byte.
How the entries of S-box come.

Book: Speech & Language Processing: Daniel Guafki

Chapter 3 Traditional Symmetric-Key Ciphers

Symmetric key = Private key = Secret key

- Singel key for both Encryption & Decryption
- Public Key has 2 set of keys
  - Encryption
  - Decryption

Cryptanalysis: Science of breaking secret codes.

- Brute Force Attack
- Ciphertext-only-attack
- Statistical Attack
- Pattern Attack
- Known-Plaintext Attack
- Chosen-Plaintext Attack
- Chosen-Ciphertext Attac

Substitution Ciphers

- Monoalphabetic
  - Additive Cipher
    $C = (P + K) \bmod 26$
    $P = (C - K) \bmod 26$
- Polyalphabetic.

Eg: 3.5

Ciphertext : "UVA"

K = 1

U = 19 mod 26 = 19 = t

for K = 6

U = (20 − 6) mod 26

14 mod 26 = 14 = O

V = (21 − 6) mod 26

15 mod 26 = 15 = P

A = (0 − 6) mod 26

−6 mod 26 = 20 = U

Statistical

If I = 14 → most common char

Positional value of I = 8

" " " E = 4 [Most commonly occuring in English]

Key = 8 − 4 = 4

If V = 14 → most common char

Positional value of V = 21

" " " E = 4

Key = 21 − 4 = 17

Mathematics of Cryptography

Finite Fields are the Galois field. It has p^n elements.

For GF(2)

Addition → XOR
Multiplication → AND

For GF(5) : find additive & Multiplicative inverse

Eg:- 4.14

Polynomial representation of a 8-bit word

1 0 0 1 1 0 0 1

Polynomial = $x^7 + x^4 + x^3 + 1$

* $(x^5 + x^2 + x) \bmod (x^2 + 1)$
* $(100110) \bmod (101)$
* $(1001)^{-1} \bmod (101)$
* $(x^5 + x^2 + 1)^{-1} \bmod (x^3 + 1)$

Operations:

Degree
1
2
3
4
5

Eg:- 4.17

$(x^5 +$

Eg:- 4.18 :
add

Additi

Multiplicatio

$P_1 \otimes P_2 =$

$P_1 \otimes P_2 =$

Multiplicat

Multiplicati

An effi

List of Irreducible Polynomials

Degree	Irreducible Polynomials
1	
2	
3	
4	
5	

Eg:- 4.17

$$(x^5+x^2+x) \oplus (x^3+x^2+1)$$

Eg:- 4.18 : Additive Identity:
adding a polynomial with itself

Additive Inverse :

Multiplication:- Eg:- 4.19. $x_7+x_7=0$, $x_7-x_7=0$

$$P_1 \otimes P_2 = x^5(x^7+x^4+x^3+x^2+x)+x^2(x^7+x^4+x^3+x^2+x)$$

$$x^{12}+x^9+x^8+x^7+x^6+x^9+x^8+x^4+x^3+x^2$$
$$+x^8$$

$$P_1 \otimes P_2 = (x^{12}+x^6+x^3+x^2)$$

Multiplicative Inverse. $\boxed{(x*y) \bmod n \equiv 1}$

Multiplication Using Compiler.

An efficient algo for multiplication using polynomials

7/Feb/2023

$0 = 0 = 0 = 0 \longrightarrow 0 = (0000)$

$g^0 = g^0 = g^0 = g^0 \longrightarrow g^0 = (0001)$

$g^1 = g^1 = g^1 = g^1 \longrightarrow g^1 = (0010)$

$g^2 = g^2 = g^2 = g^2 \longrightarrow (g+1) = (0011)$

$f(g) = g^{10} + g + 2.$ $\quad g^{10}, g^{11}, g^{14}$

$g^5 \rightarrow g^2 + g$

$g(g^2 + g + 1) = g^{10}$

$g(0011 + 0010 + 0001)$

$$\begin{array}{r} 0011 \\ 0010 \\ \hline 0001 \\ 0001 \\ \hline 0 \end{array}$$

$f(g) = g^4 + g + 1$

$0 = g^4 + g + 1$

$-g^4 = g + 1$

(-) doesn't mean anything. Can be removed

Inverses

Additive Inverse:

$-g^3 = g^3$

Multiplicative Inverse:

$(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1$

$= 1111$

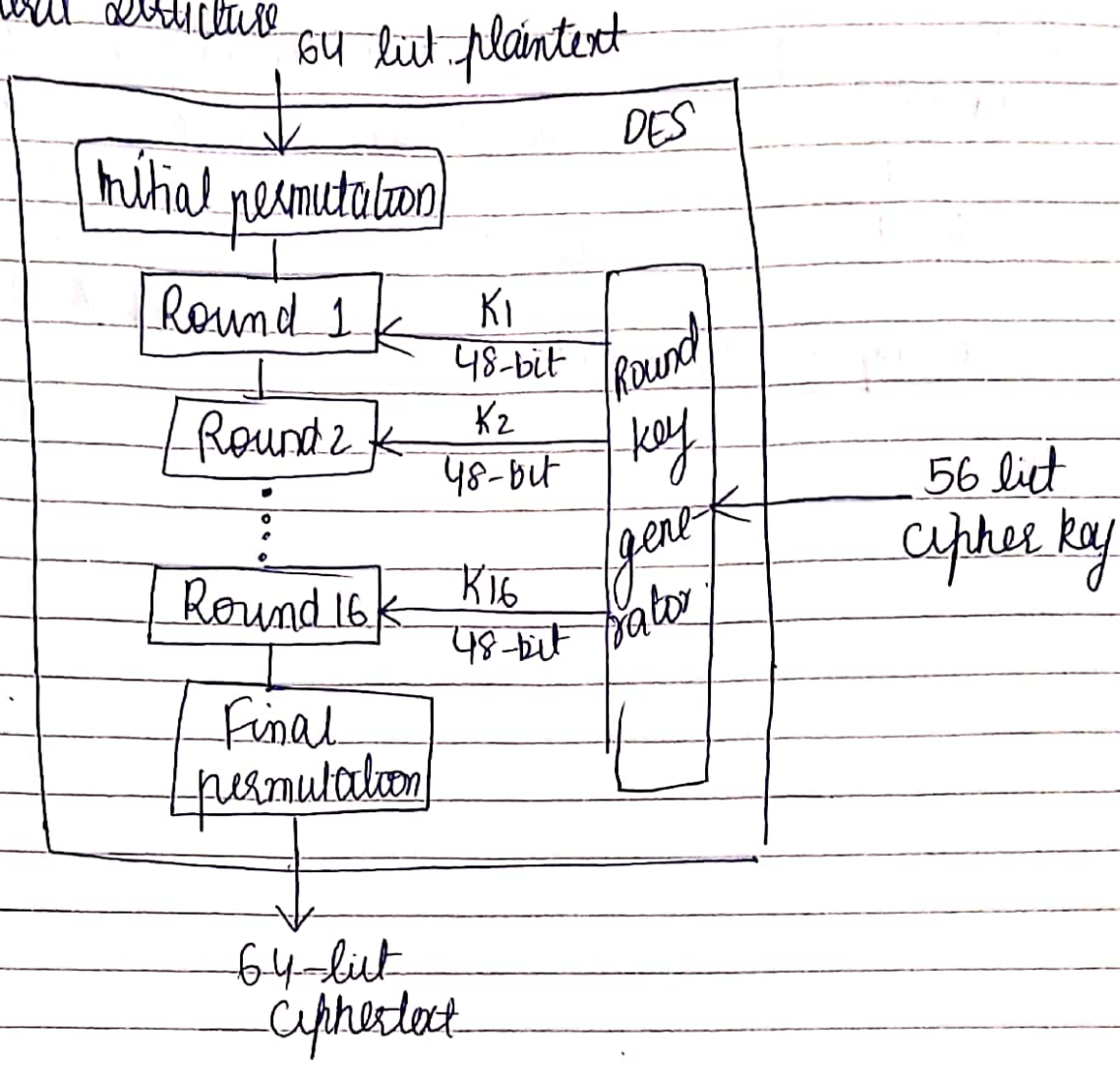
$-3 \bmod 15 = 12 \bmod 15$

g^3 and g^{12} are inverses of each other.

Eg 14-26

Data Encryption Standard (DES)

General structure



① If MSB = 0, left shift previous result by 1

② If MSB = 1, left shift & XOR MSB.

4.23 $P_1 = 00100110$, $P_2 = 10011110$, modulus = 100011010

Powers — Shift-Left Operation — X-OR

$x^0 \otimes P_2$ — 10011110

$x^1 \otimes P_2$ — 00111100 — 00111100

00001101 0

Using a Generator:

Eg: 4.24

→ Tokenization, d
→ Issues with To

→ Phonology : Pa
→ Lexicology : Com
→ Giv

→ Morphology :

Roo

→ Sentence Seg
→ Word toke

→ Porter's Algo
→ Similarity

→ Why perfo

Actual wor
many 'mis
form.

→ Given a
learning

6/02/23

NLP

→ Tokenization, Segmentation
→ Issues with Tokenization

→ Phonology : Pronunciation of word
→ Lexicology : Lexeme is an entity that combines different forms of a word.
→ Give more info. about the word.

→ Morphology : Derivational
Inflexional

Root form (play) → Inflected form (played, playing).

→ Sentence Segmentation : Maximum Matching Algorithm
→ Word tokenization : delimiter is 'space'.

→ Porter's Algorithm.
→ Similarity Algorithm

→ Why perform Normalization: ? Bcs 'gr8' is not present in the dictionary. It is a mistaken form or short form. (gr8 → great)
Actual words can have many 'mistaken' or 'short' form.

→ Given a text, we can make it a machine learning problem, binary classification problem.

- Wikipedia text is a formal text, requires least preprocessing

→ Introduction part from book

ChatGPT → NLP + NLU

Language Model: is only responsible for generating text based on existing text. This is called Corpus based NLP.

→ It does not perform understanding of text.

Model: is a funxn that processes input & produces output.

Language Model: can generate representation of language which can be utilized in further tasks.

: ML is purely based on numbers.
: Prerequisite for any ML model is that the input should be in the form of nos.
: It is then decoded into real data.

Easiest kind of language model: Assign ~~Rule~~ code to each word.

Another kind: Match the sparse matrix code with each word.

7/02/23.

Language Modelling : converts text into nos. or probabilities.

→ Probabilistic Language Model
→ N-grams
→ Joint Probability is used to calculate sentence Probability
→ Probability of words

(JP) Joint Probability : $P(AB) = P(A) \cdot P(B)$

→ Conditional Probability (CP), $P(A|B) = \frac{P(A,B)}{P(B)}$.

Sentence Probability is useful in file Summarization, Translation, Ques; Ans Generation.

CP is used in text analysis :

P("the" | "This is") : Means given that there is a sentence "This is", which word would follow.
P("that" | "This is")

"I want the probability of a word given a sequence of words"

Application of Probabilistic Language Model.
→ Machine Translation
→ Spell Correction
→ Speech Recognition

$P(Word) = P(w_1, w_2, w_3 \ldots w_n)$

How to compute Joint Probability:

P(its, water, is, so, transparent, that)

Chain Rule of Probability.

P(its) · P(water|its) · P(is|its water) · P(so|its water is)

Markov assumption

→ Bi-gram probability : Bigram model {grams = words}
→ Trigram "

Eg:- Estimating Bigram Probabilities

$c(w_{j-1}, w_i)$

<S> I am Sam </S>
<S> Sam I am </S>
<S> I do not like green eggs and ham </S>

P(<S>) → Unigram
P(I | <S>) → Bigram
P(am | <S>, I) → Trigram

Evaluation & Perplexity

The Shannon Visualization Method

Text Classification : input : a document
classes

output : predicted class.

Methods : Hand-coded rules.
Supervised Machine Learning : Naive Bayes
Logistic Regression
SVM
K-NN.

Naive Bayes

Bag of Words Representation

Multinomial Naive Bayes Classifiers to T.C

Naive Bayes Classifier

- Conditional Independence Theorem : $P(X|Y,Z) = P(X|Z)$

Y is eliminated using Conditional Independence.

$$P(X|C) = P(X_1, X_2, X_3 \ldots X_n | C)$$

$$= P(X_1|C) \cdot P(X_2|C) \cdot P(X_3|C) \ldots P(X_n|C)$$