Theory 1: Databases

Table contains
- ↳ Fields (Column)
- ↳ Records (Row)
- ↳ Attributes (Row Title)

Redundancy
- ↳ Same data being stored more than 1x
- ↳ Can lead to data inconsistency

Relational DB Format: Table (Attr1, Attr2 ...)
1. Underline Primary key
2. Dot-underline Foreign key

rDB key ⇒ Column that uniquely identifies row
- ↳ Candidate key ⇒ Any Col that Identifies specific row
- ↳ Primary key ⇒ Set of Cols that Identifies specific row
- ↳ Secondary key ⇒ Any Candidate key that isn't Primary
- ↳ Foreign key ⇒ Attribute in table referring to Pritky in other table

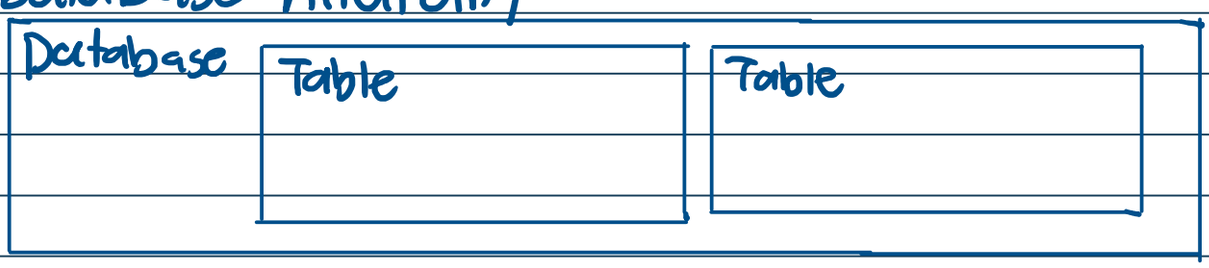
Normalisation
- 1NF ⇒ Atomicity
- 2NF ⇒ 1NF + No Partial Dependencies
- 3NF ⇒ 2NF + No Transitive Dependencies

ER Diagram
- — one to one
- >— many to one
- >—< many to many

CRUD Operations
- ↳ Create
- ↳ Read
- ↳ Update
- ↳ Delete

Database Operations
- ↳ Data Definition Language
- ↳ Data Manipulation Language
- ↳ Data Control Language
- ↳ Transaction Control Language

Database Anatomy



Theory II : Non-relational Database

Types of nrDB
- Value DB
- Document DB
- Wide-Column DB
- Graph DB

ACID Properties
- Atomicity
- Consistency
- Isolation
- Durability

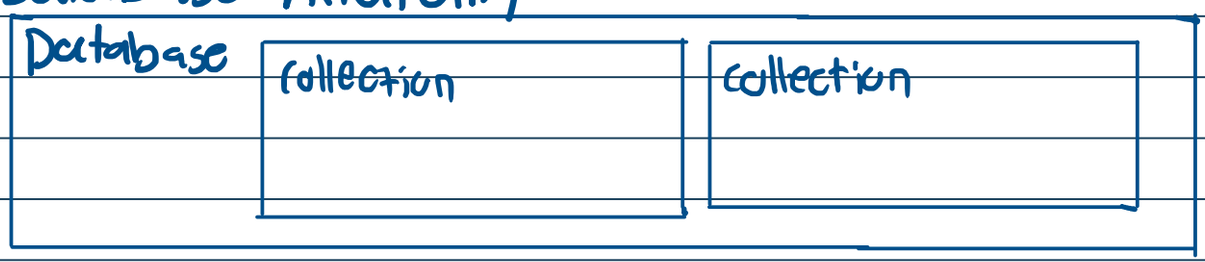
rDB Contains:
- Documents → Row
- Fields → Column

} JSON-like

nrDB Advantages
1. No Predefined Schema
2. Heirarchical Storage
3. Horizontal Scaling
4. Storage in many servers

rDB Advantages
1. Structured Data
2. Complex Querying
3. Data Integrity

Database Anatomy



Theory III: Web Apps

Types of Apps

Native App ⇒ Installed onto device
Web App ⇒ Internet enabled, needs browser
Hybrid App ⇒ Native wrapper, internet use
Progressive App ⇒ Can be made into shortcut and offline

Native Apps

1. Developed for specific OS
2. Need to be installed
3. Have access to GPS, cam
4. No need internet
5. Faster
6. Safer
7. Maintenance and Update costs

Web Apps

1. Just need web browser
2. No need to install
3. No access to GPS, cam etc
4. Need internet
5. Slow due to connection
6. Less safe
7. Updated automatically

Jakob Nielsen's Usability Principles

C → Consistency
H → Help & Documentation
A → Aesthetics
R → Recognise don't recall
M → Match to real world
F → Flexibility
U → User Freedom
V → Visibility of system status
E → Error Recovery
E → Error Prevention

Character Sets

ASCII → 128 Distinct Characters (0-31 Non-Printable)
↳ Exactly 7 bits long
Unicode → 155,000 Distinct Characters (Foreign Chars too)
↳ 8-32 bits long

Theory IV : Functions & Recursion

Subroutine
↳ Functions (has return value)
↳ Procedures (no return value)

Definitions
```
def do_something(number, letter):
```
Initialiser ↑↓ Parameters

```
do_something(13, 'R')
```
Arguments

Variables
Local → Defined within a structure or function
Global → Defined outside a structure or function

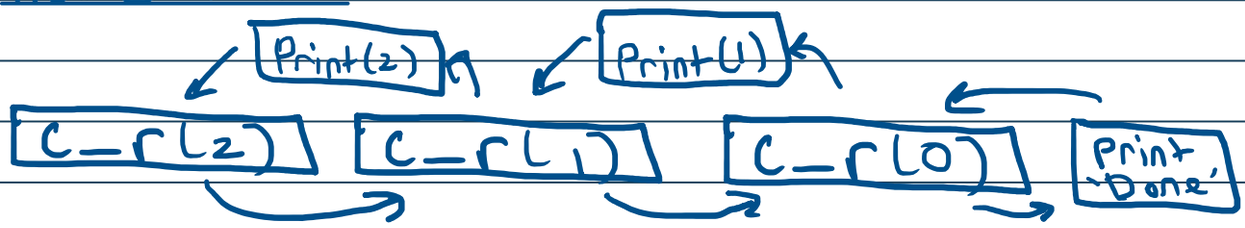
Recursion
① Defined within itself
② Calls itself
③ Terminates when it hits base case

Trace Table

call no.	call	n==0	output
1	c_r(2)	False	
2	c_r(1)	False	
3	c_r(0)	True	"Done"
2	c_r(1)	False	1
1	c_r(2)	False	2

Executions
uses stack frames to push and execute commands until base case. Then, popped from stack.

Trace Tree



Theory V: Algorithms

Sorting Algo

Bubble Sort → $O(n^2)$ worst case
1. can be optimised using swapped
2. Iterates i and iterates j until i

Insertion Sort → $O(n^2)$ worst case
1. Holds swappable element and shifts it to position
2. Best case it makes no comparisons

Quick sort → $O(n^2)$ worst case
1. can be inplace or out of place (recursive)
2. Reconstructs array based on pointer
3. If nearly sorted, fewer comparisons

Merge sort → $O(n \log n)$ worst case
1. Divide and conquer (Recursive)
2. Splits array into elements and merges to reconstruct
3. Always takes same time no matter sortedness

Search Algo

Linear Search → $O(n)$ worst case
1. Iteration
2. Can work in messy unsorted datasets

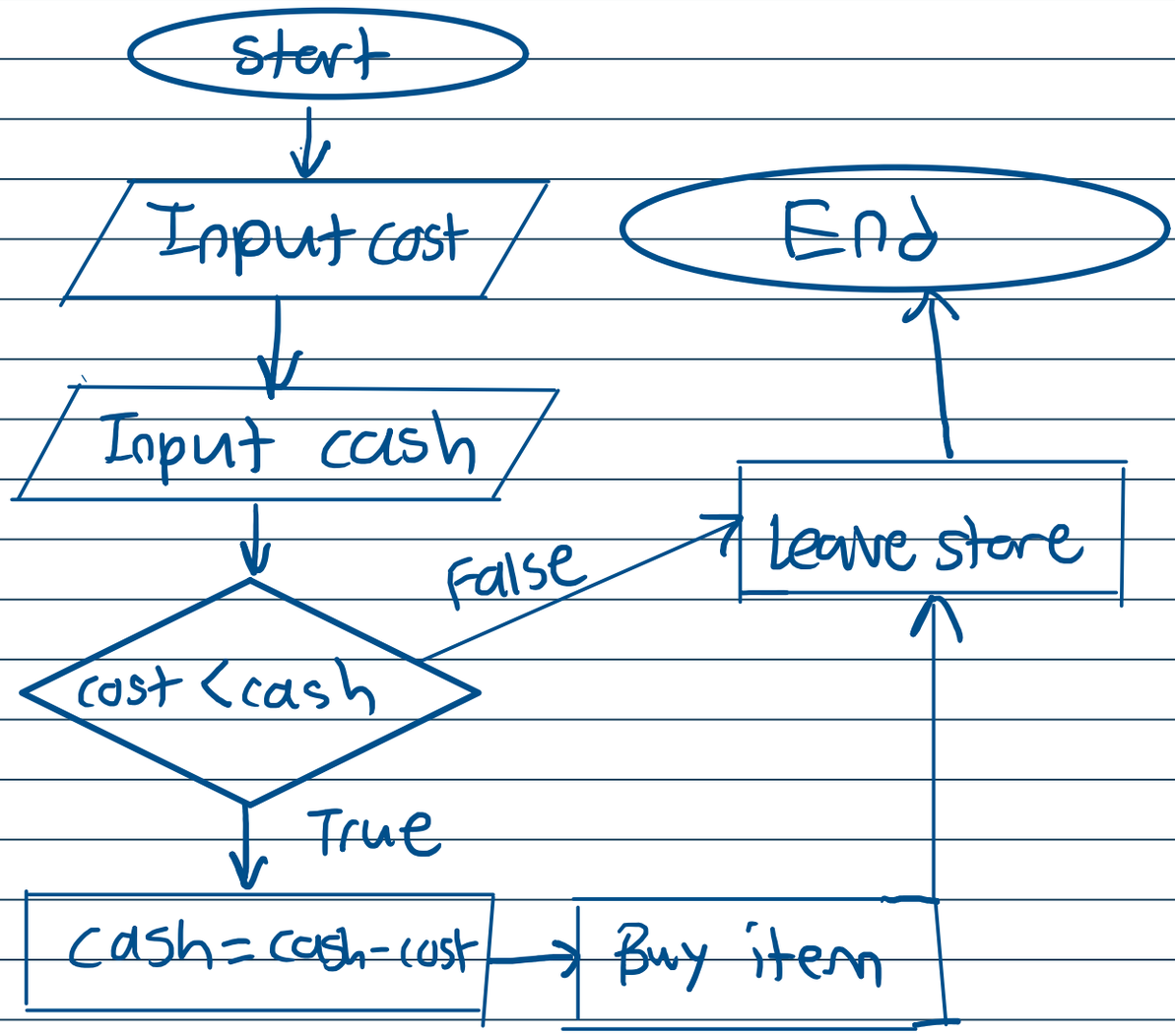
Binary Search → $O(\log n)$ worst case
1. Requires sorted dataset
2. Cuts search window in half every search
3. Repeats until left ptr crosses right ptr
4. Significantly faster in larger datasets
5. Due to halving search window, $O(\log_2 n)$

Theory VI: Control Flow

Decision Table

	Decision Table								
Conditions	Condt 1	Y	Y	Y	Y	N	N	N	N
	condt 2	Y	Y	N	N	Y	Y	N	N
	condt 3	Y	N	Y	N	Y	N	Y	N
Actions	Action 1	X				X			
	Action 2		X	X			X		
	Action 3				X			X	X

Flowchart



Theory VII: Storage/Memory

Types of Files
1. Serial Files → Records have no defined order
2. Sequential Files → Sorted in order based on key field

File Paths
1. Absolute Path → Exact file location in server/com
2. Relative Path → Location of file w.r.t current file

Shallow Copy
↳ Only copies memory value
↳ Mirrors memory location

Deep Copy
↳ Creates new data
↳ Not related to original data

Advantages of Static Memory → Stack Based
1. Faster Access as memory allocated at compilation
2. Prevents memory fragmentation due to fixed size
3. Improved Cache Performance, data in similar location

Advantages of Dynamic Memory → Heap Based
1. Flexible memory reallocation and reusability
2. Scalable as data need not be predefined
3. Safer for larger data structures, avoids overflows

Theory VIII: Data Validation & Verification

Input Errors
- Transcription Error (Hello → Heloo)
- Transposition Error (Hello → Helol)

Validation
- Ensure that data satisfies a set of rules
- Data must be Sensible, Complete, within Bounds

1. Range Check
2. Format Check
3. Length Check
4. Presence Check
5. Check Digit

Verification
- Ensuring that entered data was intended

1. Proofreading
2. Double Entry

Test Data $0<x<9$
1. Normal → 2, 3, 4, 5, 6, 7
2. Boundary → 1, 8
3. Abnormal → 0, 10, "Hi"

CheckDigit

Creation →

3 1 4 1 5 9

$3_7\ 1_6\ 4_5\ 1_4\ 5_3\ 9_2$

$84 = 21+6+20+4+15+18$

$(11 - 84 \bmod 11) \bmod 11 = 4$

∴ 3 1 4 1 5 9 4

Use →

3 1 4 1 5 9 4

$3_7\ 1_6\ 4_5\ 1_4\ 5_3\ 9_2\ 4_1$

$88 = 21+6+20+4+15+18+4$

$(11 - 88 \bmod 11) \bmod 11 = 0$ ✓

Errors
Syntax Error → wrong grammar, WILL NOT RUN

Logical Error → Incorrect Output due to logical error, WILL STILL RUN

Runtime Error → Valid Syntax and Logic, enviroment error potentially recursion, WILL CRASH

Theory IX : Object Oriented Programming

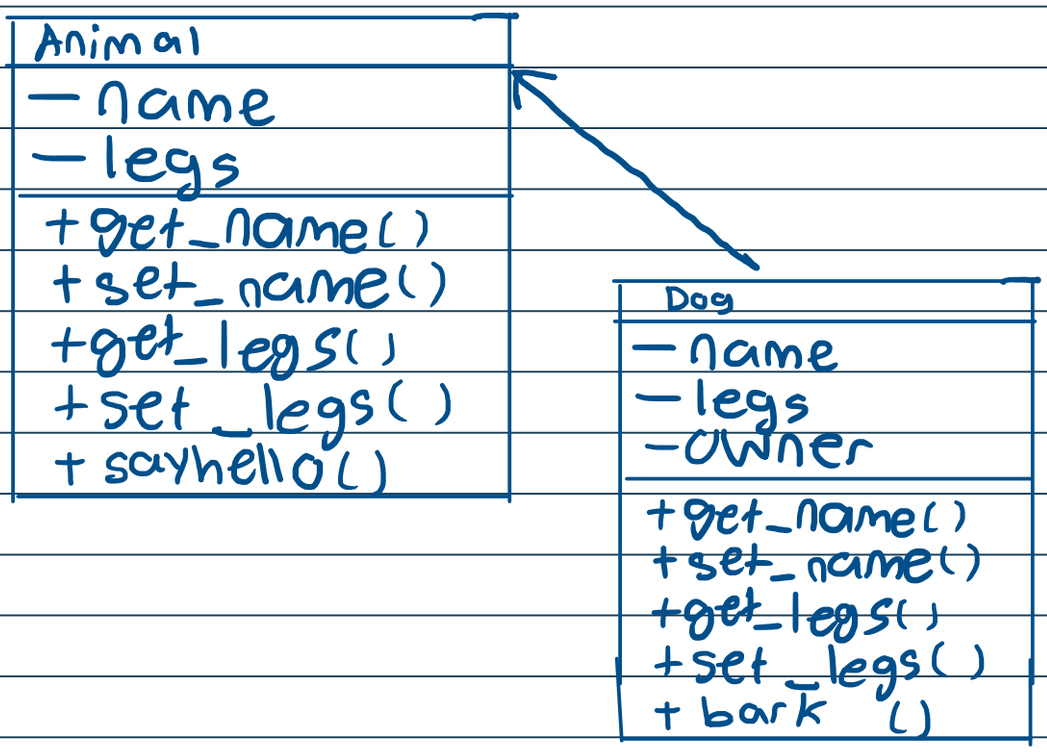
Classes

Class ⇒ Blueprint for an object
Object ⇒ Instantiation of class

Mechanism

① Encapsulation
⇒ Bundling of data and methods within a class
↳ Restricting access to private attributes with public methods
↳ Seperates interface from implementation
② Inheritance
⇒ Enables child classes to access public methods from parents
↳ Promotes code reusability
↳ Reduce code duplication
③ Polymorphism
⇒ The redefinition of a method in a child class with different behaviour
↳ Promotes code generalisation
↳ Reduces complexity of code

Class Diagram



Theory X: Data Management & Ethics

Data ⇒ Information encoded and structured for processing
Information ⇒ Data that has been interpreted
Data Integrity → Ensures ① completeness ② consistency ③ validity
Data Privacy ⇒ Ensures that Data is disclosed to authorized people only
Data Life Cycle ① Creation ② Storage ③ Usage ④ Archival ⑤ Deletion

Storage Threats
① Human Behaviour
② Overwriting
③ Unauthorized Intrusion
④ Natural Disaster

Code of Conduct
↳ Competence
↳ Responsibility
↳ Integrity
↳ Professionalism

NRIC
→ Illegal to collect unless to verify ID
→ Partial NRIC used otherwise

Legislation: PDPA, CybersecurityAct, Computer Misuse Act, Copyright Act

Protection of Data
① Backup → Create copies of data as safety precaution
② Archival → Storing historical data for auditing purposes
③ Access Restriction → Passwords, 2FA, IAM, RBAC
④ Version Control → Git {modify, stage, commit}
⑤ Validation & Verification *
⑥ Encryption → Alter data using key to make it unreadable
⑦ Erasure → Completely overwrite old data to limit liability

PDPA Obligations: Rules governing Collection, Use, Disclosure of Data
① Purpose Limitation → Disclose uses of collected data
② Accuracy → All collected data must be accurate & complete
③ Protection → All data must be securely protected
④ Accountability → Make info on protection policies available if asked
⑤ Consent → Ask for consent before using / disclosing data
⑥ Access & Correction → If someone requests their data, give it back in 365d
⑦ Notification → Inform users before collecting / using / disclosing
⑧ Transfer Limitation → Make sure data is legal when brought overseas
⑨ Retention Limitation → Reduce users' tie to their data for easy deletion

Theory XI : Networks A

Computer Networks

Network • System of >2 computers connected by transmission medium
LAN • Network of Devices in small Geo Location
MAN • Network of Devices in larger Geo Area
WAN • Network of multiple MANS/LANS across countries/continents
Intranet • Private Network built into org for secure info sharing amongst authorised users
Internet • Global public Network accessible to anyone with connection

Network Hardware

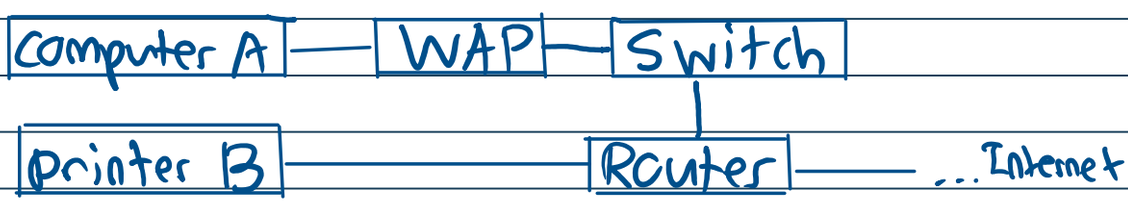
NIC • Chip has MAC address, allows wireless transfer b/w device and network
UTP Cable • Standard cable for electrical transmission
Hub • Connects devices within network, manage data flow, shouts recieved packets
Switch • Connects devices within network, manage data flow, logs MAC addresses
Router • Connects to LANs, connects to networks, assign IPs, direct packets
WAP • Devices that create WLANs
Modem • Facilitates signal conversion and reversion

Network Addressing

MAC • Static, 48b, Given by NIC chip — 1A:4C:...:3C (×6)
IPV4 • Dynamic, 32b, Given by router to Device, 255.255.255.255
IPV6 • Dynamic, 128b, Given by router to Device, 14A6:...:1376 (×8)

TCP/IP Model

Physical • Handles physical transmission of Data over wires
DataLink • Packages data into frames for devices in network segment (NIC, switch)
Internet • Routes packets across different networks (Routers)
Transport • Manages data transfer between applications (TCP/UDP)
Application • Provides services directly to apps (Apps, OS, Browser)



Theory XII: Networks B

Transmission Protocols

1. Hypertext Transfer Protocol • Requests webpage, returns 200 ✓ 404 X
2. File Transfer Protocol • Used to upload / download files
3. Internet Message Access Protocol • Synchronises emails with server
4. Simple Mail Transfer Protocol • Used to send email to email server
5. Dynamic Host Configuration Protocol • Used by routers to assign IPs
6. Transmission Control Protocol • Reliable exchange between Apps
   ↳ Data split into Packets, arranged at end, missing are resent
7. User Datagram Protocol • Fast exchange between Apps
   ↳ Data split into Packets, arranged at end, missing are ignored
8. Internet Protocol • Packages data with host IP & Destination IP

Get: Allows of only retrieving data **Post**: Allows CRUD of server data

DNS Request Process ⇒ Domain Name Service (Make IP into .com)

1. DNS Cache Checked. If not found, request sent to recursive DNS server (ISP)
2. rDNS server Checked. If not found, request sent to root DNS servers
3. Root Server checks TLD (.com), refers to correct TLD server
4. TLD Server refers to Authoritative Server, request processed
5. Response travels back, caching at every cache for future use

Process Network Interface

Socket • Software interface where data is exchanged between process & server
- ↳ Interface bw Application & Transport Layer
- ↳ Protocols (TCP/IP) reduce lag and data chop-ups
- ↳ Bidirectional, needs IP and port no (1024-65535)

Client Server Architecture

1. Centralised Management
2. Reliability
3. Security
4. Scalability

P2P Architecture

1. Cost Efficiency
2. Availability if one connection dies
3. Load Distribution
4. Decentralisation

Theory XIII: Transmission & Switching

Packet switching

↳ Resources not reserved (Transmission Rate / Buffers)
↳ If congested, packets are buffered
↳ The internet is a PS based Network
↳ **Store and Forward** : Each switch waits for all packets before forwarding
↳ **Packet Loss** : Packets dropped if buffers are full
↳ **Congestion** : Reroutes buffer packets if recieved faster than processable
↳ **Lost Packages** : If destination unreachable, packets are sent to a default device, looping until TTL expires
↳ **Forwarding Tables** : IP Addresses in packet headers for forwarding

Circuit Switching

↳ Resources reserved for full time (Transmission Rate / Buffers)
↳ Multiple switches have connections in and out
↳ Transmission Rate is shared amongst active connections

Advantages of CS

1. Dedicated Connection
   ↳ Constant and predictable
2. Speed
   ↳ No packet queue
3. Reliability
   ↳ Avoids packet loss
4. Long Distance
   ↳ No overhead for sending individual packets

Advantages of PS

1. Efficient use of Bandwidth
   ↳ only needed bandwidth used
2. Scalability
   ↳ can grow cheap and easily
3. Cost Efficiency
   ↳ Fewer resources used
4. Fault Tolerance
   ↳ Any failure node can be resolved through rerouting

Theory XIV: Security

Malware

1. Ransomware
2. Scareware
3. Spyware
4. Adware
5. Fileless Malware (hides in memory)
6. Cookies
7. Pharming (Intercepts requests, redirects)
8. Phishing (Email/Website Impersonation)
9. Spamming (Unwanted messages)
10. Trojan (pretends to be harmless file)
11. Virus (Attaches to harmless program, modifies, spreads)
12. Worm (Attaches to network, consume bandwidth, spreads)

CIA Triad

- Confidentiality
- Integrity
- Availability

Denial of Service

Attempt to overwhelm website, network or server with requests to render it inaccessible or slow

DDOS ⇒ uses a botnet of slaves

Firewalls ⇒ Filter that monitors access

Types: Hardware, Software, Host-based, Cloud-based, Network based

Functions: Traffic Control, Maintaining Security

Applications:
1. Stateful Packet Filters (Track TCP Connections)
2. Packet Filters (Check Packets)
3. App Gateways (Check AppData)

Limitations:
1. Cannot protect from internal threats
2. Single Point of Failure
3. Cannot protect against threats users allowed

IDS ⇒ Device that generates alerts when detects potentially malicious traffic

Detections:
1. Signature Based - Uses database of known attacks
2. Anomaly Based - Detects unusual streams

Use cases: alerts in nMap, portscan, TCP stack scans, other attacks

DMZ:
1. High security zone (HSZ) contains WebServer, DNS server, FTP Server
2. Pfilter protects DMZ and HSZ, App Gateway protects HSZ
3. IDS monitors both, does deep packet inspection, signature comparison
4. Placing IDS downstream reduces traffic load per sensor
5. Alerts sent to admins, management systems, or logged

Theory XV : Representation & keys

Number Systems

Binary ——— 0 1 → 10 11 → 100 101 110 111
Octal ——— 0 1 2 3 4 5 6 7 → 10 11 12
Decimal ——— 0 1 2 3 4 5 6 7 8 9 → 10
Hexadecimal - 0 1 2 3 4 5 6 7 8 9 A B C D E F

Conversion

Decimal value of 1011 : $[1 \times (2^3)] + [0 \times (2^2)] + [1 \times (2^1)] + [1 \times (2^0)]$
$= 8 + 2 + 1 = 11$

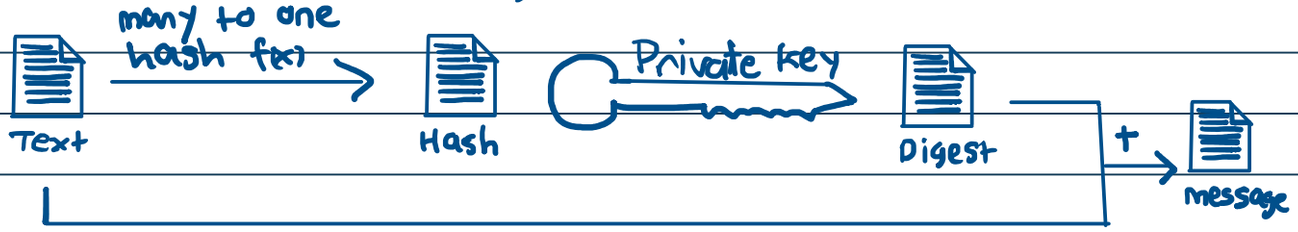
Binary value of 6 :
6 / 2 = 3 R0
3 / 2 = 1 R1
1 / 2 = 0 R1
Reverse to form: 1 1 0

Symmetric key Encryption

Asymmetric key Encryption

Digital Signature → For non-repudiation

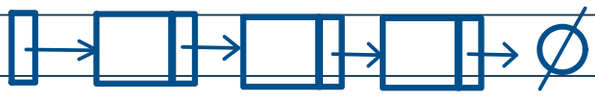
① Sender uses one-way hash function to create digest
② Private key is used to encrypt digest
③ Both are sent to receiver
④ Message hashed using f(x). Digest decrypted using public key.
⑤ If the two results match, sender is verified.



Theory XVI : Data Structures

Linked List

Insertion, Deletion : O(n)
Head Insertion, Deletion : O(1)
Searching : O(n)

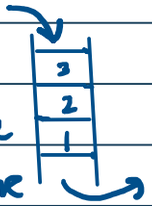


Stacks (FiLO)

insertion = push
deletion = pop

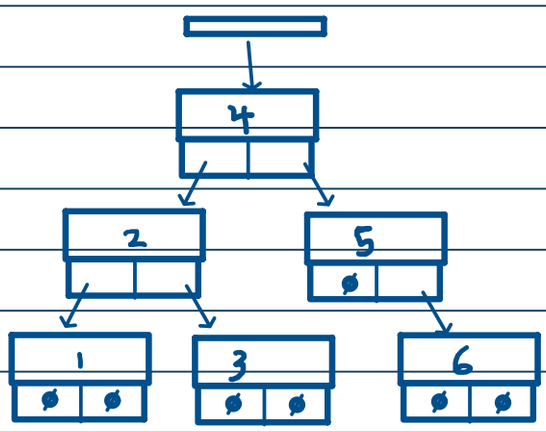
Queue (FiFo)

insertion = enqueue
deletion = dequeue



Binary Trees

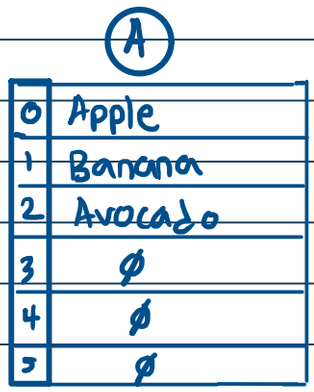
Balanced Tree
↳ Insertion, Deletion : O(log n)
↳ Searching : O(log n)
Unbalanced Tree (Linked List)
↳ Insertion, Deletion : O(n)
↳ Searching : O(n)



Hash Table ⇒ Uses Hash F(x) to index

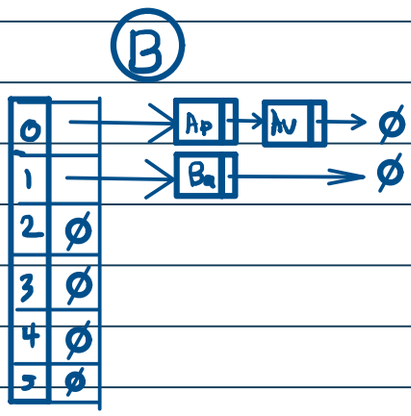
Open Address / Close Hash (A)

↳ Hashtable size = No. keys
↳ No size overhead / Memory overuse



Close Address / Open Hash (B)

↳ No issues with clustering
↳ Performs better with high load



Good Hash Table

↳ Roughly 1.5x data size

Good Hash F(x)

↳ Fast to compute
↳ Uniform Distribution