

Hisham Paloli

Theory

Javascript

- illegal shadowing
- variable shadowing
- function composition
- null pointer exception
- nullish operator
- starvation
- method chaining
- polyfills
- rest operator
- block scope
- function scope
- this keyword
- this in arrow function
- this substitution
- What is a private constructor function?
- rest operator (Use it outside a function)
- private constructor
- Function composition (what is it? And where is it used?)

Node JS

- preflight request
- path parameters
- dynamic routing
- router chaining
- promisify
- cron-job
- rate limiting (also types of rate limiting)
- write head
- set header
- i/o bound vs cpu bound
- process.nextick

- CSRF (How do malicious sites implement CSRF attacks?)
- concurrency
- Parallelism
- worker thread
- thread pool
- how node handles i/o bound vs cpu bound operations
- token introspection
- API versioning
-

Database

- Views
- materialized views
- type of data modeling
- covered query
- clustered collection
- clustered index
- batch sizing
- db profiler
- gridFs
- levels in isolation
- isolation ?
- transaction
- drawbacks for normalization
- drawbacks for indexing
- \$addFields
- \$each
- \$facet
- find the color wise avg price
- enable journaling
- sharding

React

- react drawbacks
- why react is not seo friendly
- rules of using hooks?
- babel
- webpack
- useCallback

- useMemo
- react.memo
- error boundary (How to use it with functional components?)
- native events
- synthetic events
- event pooling
- shadow DOM
- hot module replacement
- forwardRef

DSA

- degenerate tree
- monotonic stack
- monotonic queue
- separate chaining
- hashtable time complexity
- implement stack with methods push, pop and getMax. All of the should work in $O(1)$ time.
- how to minimize collision
- bloom filters
- minimal perfect hash function
- Weighted graph

Practical

- write a function to convert a obj to string

```
let obj = {
  name: 'hi',
  age: 20
}
output = "name=hi&age=20"
```
- write a function to check if a linked list is circular or not
- write a function which will check if a target value is present in an sorted 2d array.

```
let array =
[  
  [1,2,4],
```

```
[5,7,9],  
[12,34,45]  
]  
let target = 9
```

Vishnu GR

JS

- immutable datatype
- features of JS runtime
- falsy values
- mark and sweep
- String prototype, boxing
- eval()
- creating private scope
- JIT compilation
- ?? vs ??=
- 1 | 2 vs 1 || 2
- bind()
- class: private property
- shallow copy object but exclude a key
- print current time without printing date
- build object from array of keys and array of values but ignore null values

- getters and setters
- private properties
- what datatypes can be passed to JSON.stringify
- String prototype, boxing
- why eval() is considered dangerous
- "use strict"
- inheriting Array
- event delegation
- pure functions (read more)
- is JIT used all the time?

- current date in D/M/Y format
- execute a callback function every n seconds up to 10 times
- generator function that yield random unique elements
- swap keys and values of object, ignoring null values

DB

- \$addFields vs \$set
- namespace
- atomicity
- skill known most
- set hire_date to current date
- reduce salary by 10% if developer doesn't know html or css

- WiredTiger
- partition tolerance
- createView
- wildcard index
- text index
- orders with exactly two items
- orders delivered 2 days ago
- orders with 3 or more total items

Frontend

- rgba
- innerHTML vs textContent
- cookie storage
- disadvantages of localStorage
- uncontrolled components
- immutability of state
- defaultProps
- key prop
- reading query params and path params
- ESModules
- named exports

- applications of refs
- advantages of redux tk slices
 - viewport
- reading current URL
- XSS
- httpOnly cookies
- MemoryRouter
- immutability of state
- React fiber
- ESMODules
- importing named exports without destructuring
- return data of reducer function
- <a> vs Link
- hydration
- how frontend can know token expiry

Backend

- thread vs process
 - worker threads
 - running a function without setImmediate vs using it
 - exporting in CommonJS
 - socket vs socket.io
 - using environment variables
 - configuring and using path params
 - express.urlencoded
 - Express.set
 - HTTP 403, 201
 - refresh token
 - JWT claims
 - reverse proxy
 - default HTTP port 80
 - SSH
- CommonJS

- os module
- environment variables without using .env
- socket
- thread, process
- parts of HTTP response
- HTTP 401, 500
- configuring app settings
- res.locals
- JWT algorithm
- how JWT signature helps in verification
- why reverse proxy was used in hosting
- DNS

DSA

- rehashing
- backtracking
- ASCII vs UTF-8
- homogeneous
- shuffle array
- selection sort (partially done)
- Title Case string (partially done)

- advantages of static memory allocation
- remove extra whitespaces from string
- stack with ability to pop from both ends with constant time complexity

Anbin

Pendigs 1:

asynchronous programing async/await
timer functions
proxy object implementation
stopPropagation
reduce
weakset
block request based on custom header
creating child_process
clustering-need clarity
phases in eventloop
transform stream vs duplex stream
create a custom event emitter-need clarity
reactor pattern
create a ttl index
create a multikey index
create a compound index
\$addtoset
\$bulkwrite
\$inc
aggregation pipeline
\$avg
\$sum
distinct
group
find depth of bst-partial
react fiber
child to parent communication-need clarity
sibling communication
call and put in redux saga
shadow dom
mapStatetoprops
mapDispatchtoProps
render component while resizing

Pendings 2:

Deep cloning of an object
call and apply
implementation of proxy object
generator function implementation
reduce -second smallest -odd number-listen question carefully
child_process
create a child_process
worker thread
custom event emitter
transform stream
duplex stream
phases in eventloop
promise with fs
create a capped collection
project compound index
aggregation pipeline
group
distinct
\$lookup
\$bulkwrite
child to parent and parent to child communication-need clarity
react fiber
siblings communication in react
synthetic events
communication between components using use ref
Quadratic probing
double hashing
sort a stack using temporary stack

Pendings3:

Proxy object for object validation
callback with rest parameter-need practice
rate limiting techniques
every ,some

promise with includes
create endpoint for read content from file
create endpoint for write content into file
create a custom event emitter
fork vs spawn
libuv
transform stream
reactor pattern
phases in eventloop-need clarity
sort a stack using temporary stack
hashtable implementation
AVL
\$bulkwrite
Sibling communication in react
rerender component when page resized
custom hook-need practice
synthetic events
combineReducers

Rahul Ranjan

Pendings 1:

Javascript:

Weekmap vs weekset
Promise vs observables
Es6/7 features
Write a code for create a custom map or filter function
P-6, t-6

MongoDB

\$let

How to improve query performance (take more deep understanding)

T-6, P- 6

Data structure & Algorithms:

-Reverse the string to it's own place using stack

convert nested array into flat

t-5, p-5

React:

Explain optimisation features in React (Take deep and practical understandings properly)

Create nested dropdown and map the array of object by selection if 1st dropdown, update the list if nested dropdown (partial done)

t-5, p-5

Node:

What are the New features introduced in node js

XSS attack

t-6, p-6

Pendigs 2:

JavaScipt:

Weakmap vs WeakSet

ES7 features

Create callback hell convert into promise and make more efficient using
async/await

Event propogation vs event delegation

Default port of http and https

t-6, p-4

Node:

XSS Attack

how to secure node api

fork vs spawn

Create worker thread

New features in node js

cluster modules vs worker thread

Take practical understanding of worker thread

-create an Express server with protected routes using middleware that verifies authentication through request headers?

t-6.5, p-4

MongoDB

- \$map

- \$let

- \$cond

- \$facet

- Isolation Example

- Profiler

-write a query to increase salary of employee by 10% whoise age is less than 35

-Find the count of the employee whoi age is greater then equal to 20

T-, 4, P- 3

Data structure & Algorithms:

-Remove the duplicate from nested array using recursion or without

-Convert nested array into flat array using recursion or without

- Write a code to find the longest substring pelindrome in the given string

-Binary search using recursion

-Find the longest common prefix using trie

- Write a code to check if given parenthesis is balenced?

-Reverse the string to it's own place using stack

-find middle element using linklist

-Read BST, BT, Graph quetions, Heap sort

!! Added Extra question set for practice !!

t-3, p-2

React:

Difference between forwardRef and createRef?

Explain optimisation technic in React (Take practical understandings properly)

Normalisation

state vs props

redux vs context

-write a program to create circle whenever cursor will move, circle will be moved along with

t-6, p-5

Aparna

JS:

Event delegation

variable shadowing

call,apply,bind

coercion

pure function

pass by value vs pass by reference

arrow function vs regular function

remove specified index elements from array - check on splice

promise example

promise methods

object workouts

React:

one way data binding
controlled vs uncontrolled element
lifecycle of functional component
advantages of useRef
forwardRef
useMemo vs useCallback
Pure components
principles of redux
axios interceptors
React.Fragment

DS:

Divide and conquer clarity
quick sort
remove duplicates from a string - completion
reverse a stack
Check on workouts and theory

Node:

Node concurrency
router chaining
HTTP OPTION
types of middleware
query vs path params
fork vs spawn

Mongo:

clustered collection
journaling
covered query
replication
\$facet

check on practical's

Pendings -2:

JS:

- Event delegation
- illegal shadowing
- pure function - clarity
- console.log(1+ +'2')
- console.log('A'-1)
- console.log(null==undefined)
- promise.allSettled
- weakset
- weakmap
- remove second last from array
- find non repeating elements from array

React:

- passing data from parent to child
- passing data from child to parent
- reconciliation clarity
- when does a page rerender
- controlled vs uncontrolled elements
- useRef clarity
- ways to optimise react component
- lifecycle of functional component
- lifecycle phases
- core components of redux
- access token vs refresh token

DS:

- reverse a string using recursion
- quick sort implementation
- complexity of quick sort

reverse a stack

complexity of BFS and DFS of graph

Node:

Node concurrency

app.all

HTTP OPTION

router chaining

query vs path params

spawn vs fork

microtask vs macrotask queue

Mongo:

clustered collection

capped collection

journaling

covered query

\$out

\$setUnion

Shahabas

DOM manipulation methods

primitive vs non-primitive datatypes

what is promise

methods of promise

examples for HOF

null vs undefined

default parameters

localStorage vs sessionStorage

object methods

core modules in node.js

http in detail
OPTIONS
types of middleware
app.use() vs app.set()
process vs threads
delete a file using fs
embedded documents
\$all vs \$in
types of indexing
journaling
error handling in react
transpiler
pure components
render props
asymptotic notations
underflow in stack
methods to resolve hash collision
applications of heap

Swathy

Pendings 1:

NodeJS & JS

flatMap(), Symbol()
limitations of closure
arrow function - hoisting
Double exclamation,
const name = res ?? 'fault';
Hashing, encryption
event module, event.on, event.emit
setHeader, writeHead
absolute path, relative path
Express middleware

app.use(), express.urlencoded
express.static(), public
CORS, preflight, CSRF, XSS
OPTIONS
Put, patch - 100, 200, 401

MongoDB

createCollection
Insert many - array of docs
renameCollection
\$upsert
Create capped collection
isCapped, TTL index
Mongodumb, mongorestore
Utility commands
Replica set, bulkWrite

ReactJs

Debouncing in input - need clarity
useLayoutEffect, useEffect
Creating custom hook
Lifecycle methods
Polifills, networking - axios, fetch
React fiber, react profilers,
react fragment
useRef usage
Creating a timer

DSA

Linked list chaining
Probing
Collision control
Spanning tree

hashmap

Pendings 2:

NodeJS & JS

Array.some() vs Array.every()
String(), toString(), lamda expression
Symbol(), flatMap()
const {name = "abcd", age = 0} = res;
unary function, Object.entries(obj);
__dirname - absolute path
encode a url - need more clarity
Child process - spawn, fork

MongoDB

TTL index, expireAfterSeconds
\$out, \$upsert
mongosh - utility
Mongodumb backup

ReactJs

cloneElement
No dependency array useEffect
Updating ref - useRef
Pure component - in class
react fragment vs div,
useEffect - return

DSA

comparison-based and non-comparison-based sorting
Prefix tree

Pranav

js

create callback hell

use promise to solve that

create weakMap and WeakSet

Node

write 2 files, and read both to save it to the 3rd file, and show the result in page

React

useCallback

useMemo

show the difference before and after while using useMemo

DSA

find the longest consecutive repeating characters in string

Input: tryurrrrrrtttrr

output should be rrrrr

find the second longest word in sentence without inbuilt split function

input: my name is Rahul

output should be name

MongoDb:

delete one field from all the data

Highest employee joined year

Vipin Varghese

Week 21: Full in Review	Week Repeat	<p>JS : Pr:1,Th:4.5 - array structure - currying - arrow fn vs regular fn(this) - strictmode</p> <p>DS - 4 - DFS - BFS - Max heap</p> <p>Mongodb - 4.5 - replication - disadvantages of index</p> <p>Nodejs - 4.5 - concurrency vs parallelism - cors(lacks clarity) - unlink (lacks clarity)</p> <p>ReactJS - 4 - webpack - redux saga - axios interceptors (lacks clarity) - useeffect cleanup</p> <p>Theory: 4 Practical: 4</p>
-------------------------	-------------	--

Practicals Questions

1. const ab= [1,3,9,45,123,12,112,123,12,-1,-2,-3]

// 9 + 3 + 1 = 13

// 12 + 123+ 45 = ...

// 12+123 + 112 = ..

2. 100 98 94 88 80 70 58 44 28 10

// Print elements in this order

3. Sum of middle 5 elements in linked list
4. Merge Sort
5. Quick Sort
6. const ab = [{a:[34,3433]}, {a:[34,3433]}, {a:[334,3431]}]
 // count number of 3 in the array of objects

7. Common Elements in two array Using hash table.

8. let arr = [[1],[4,2],[6,1,3],[8,2,1,4]]

// Print 1 2 3 4

Akshaya Moorthy

Pending :

Conditional rendering
File upload front end
Protected route
Dynamic route
Security issues of local storage
Event delegation
Issue of react memk
Multi key vs Compound index
Lazy loading
Nodejs Cluster
Create child process
Hash map vs hash table
Complexity analysis
Space complexity of quick sort and Merge sort
Insertion
Bst search complexity

Bst Deletion
Custom hooks
Memory leaks in React
Batching
Pagination
Micro task queue and macro task queue
Tree
Cache vs buffer
Recursion to check prime
Setimmediate vs process.next Tick
Bst
Heap
Covered query
Issue with ht
Cap theorem

Jeswin

Pendings :

Generator function to generate multiples of given input number

Concurrency in nodejs - learn more
exec
execFile
app.locals
fs.link
fs.unlink
Content negotiation

Axios interceptors
Axios canceltoken
Event pooling
React fiber
Shadow dom

Html sanitization

Uncontrolled components- learn more

Hashed index

Create compound index

Database profiler

Gridfs

Aggregate practicals

Sparse array

Jagged array

Load factor

Linear probing

Quadratic probing

Segment tree

Red black tree

Minimum spanning tree

Jobin George

- Finally syntax
- Error objects - check it with instanceof
- throws statement
- Pure functions
- Generator function - first n even numbers
- Callbacks
- Promise - resolve, reject
- IIFE benefits
- Closure benefits
- Inheritance in JS

- Rest operator
- Bind concept and code
- NodeJS - what is it?
- Thread pool
- Process vs thread
- Design patterns - MVC
- fs.existsSync()
- res.render()
- \$addToSet
- \$inc decrement queries
- \$sum
- TTL index
- Hook concept
- useDispatch hook
- useCallback scenario
- JWT.verify()
- Divide and conquer
- Shortest distance algorithm
- DFS - backtracking
- Complete binary tree example

Pranav Sanker

Pendings:

- advantage of closures
- difference between fs.readFile() and fs.createReadStream()
- why let can't be redeclared when var can be redeclared
- difference between useEffect and useLayoutEffect
- useMemo
- purpose of middleware in redux
- purpose of middleware in redux
- action in redux

- `$let`
- remove the middle element from array
- time complexity of quick sort
- create a function that time complexity should be $O(n \log n)$
- find the longest consecutive repeating characters in string
- write a callback hell
- implement weakset and weekmap
- workerthread
- show an example for pure and impure components
- useCallback