

CS301P FINAL TERM PREPARATION

صرف ایک ویڈیو اور۔۔۔
CS301P کی تیاری مکمل FINALTERM

CS301P Final Term Preparation
Complete in one video



📌 Lab 1: Linked List

Q1. Linked List کس میں ہوتا ہے؟ data اور next pointer میں

- A) List
- B) Node ✓
- C) Faculty
- D) Main

Explanation: Linked list رکھتا ہے۔ address کا next node اور data node کا ہر

Q2. Linked List linear data structure ہے یا non-linear؟

- A) Non-linear
- B) Tree
- C) Linear ✓
- D) Graph

Explanation: linear ہوتے ہیں، اس لیے linked میں nodes sequence ہے۔

Q3. Head pointer کیا store کرتا ہے؟

- A) Last node
- B) Middle node
- C) First node address ✓
- D) Null

Explanation: Head point کرتا ہے۔ first node کو ہمیشہ

Q4. Empty linked list میں head کی value کیا ہوتی ہے؟

- A) 0
- B) Garbage

- C) Random
- D) NULL ✓

Explanation: Empty list میں head NULL ہوتا ہے۔

Q5. Linked list کا main فائدہ کیا ہے؟

- A) Fast access
- B) Fixed size
- C) Dynamic size ✓
- D) Sorting

Explanation: Linked list runtime پر grow/shrink ہو سکتی ہے۔

□ Lab 2: Stack (Array & Linked List)

Q1. Stack پر کام کرنے کا principle کس پر

- A) FIFO
- B) LIFO ✓
- C) FILO
- D) Random

Explanation: Last inserted element پہلے سے سب سے پہلے remove ہوتا ہے۔

Q2. Stack میں insertion operation کیا کہلاتا ہے؟

- A) Delete
- B) Pop
- C) Push ✓
- D) Peek

Explanation: Push element add کرنا ہے۔

Q3. Stack overflow کب ہوتا ہے؟

- A) Stack empty ہو
- B) Stack full ہو اور push کریں ✓
- C) Pop کریں
- D) Stack delete ہو

Explanation: Full stack میں insertion overflow کہلاتا ہے۔

Q4. Stack underflow کب ہوتا ہے؟

- A) Stack full ہو
- B) Stack empty ہو اور pop کریں ✓
- C) Push کریں
- D) Display کریں

Explanation: Empty stack سے pop underflow ہے۔

Q5. Linked list stack کا فائدہ؟

- A) Fixed size
- B) Extra memory

C) Dynamic size ✓

D) Slow

Explanation: Linked list stack memory dynamically use کرتا ہے۔

□ Lab 3: Queue (Linked List)

Q1. Queue کس principle پر کام کرتی ہے؟

A) LIFO

B) FILO

C) FIFO ✓

D) Random

Explanation: First come, first served-

Q2. Queue میں insertion کہاں ہوتا ہے؟

A) Front

B) Middle

C) Rear ✓

D) Anywhere

Explanation: Enqueue rear پر ہوتا ہے۔

Q3. Queue میں deletion کہاں سے ہوتی ہے؟

A) Rear

B) Middle

C) Front ✓

D) Random

Explanation: Dequeue front سے ہوتا ہے۔

Q4. Empty queue condition کیا ہے؟

A) front != NULL

B) rear != NULL

C) front == NULL ✓

D) size > 0

Explanation: Front NULL کا مطلب ہے queue empty ہونے کا مطلب

Q5. Queue کا real-life example؟

A) Stack of plates

B) CPU recursion

C) Printer queue ✓

D) Tree

Explanation: Printer jobs FIFO order میں process ہوتے ہیں۔

□ Lab 4: Binary Search Tree (BST)

Q1. BST میں left child کی value ہوتی ہے؟

- A) Greater
- B) Equal
- C) Smaller ✓
- D) Random

Explanation: Left subtree میں smaller values ہوتی ہیں۔

Q2. BST میں right child کی value ہوتی ہے؟

- A) Smaller
- B) Equal
- C) Random
- D) Greater ✓

Explanation: Right subtree میں greater values ہوتی ہیں۔

Q3. Duplicate values BST میں؟

- A) Allowed
- B) Always inserted
- C) Ignored
- D) Not allowed ✓

Explanation: Standard BST duplicates allow نہیں کرتا۔

Q4. BST search complexity (average case)؟

- A) $O(n)$
- B) $O(\log n)$ ✓
- C) $O(n^2)$
- D) $O(1)$

Explanation: Balanced BST میں search $\log n$ ہوتا ہے۔

Q5. Minimum value BST میں کہاں ملتی ہے؟

- A) Rightmost
- B) Root
- C) Leftmost ✓
- D) Leaf

Explanation: Leftmost node minimum value رکھتا ہے۔

□ Lab 5: BST Traversals

Q1. Inorder traversal BST show میں کیا کرتا ہے؟

- A) Random order
- B) Reverse order
- C) Sorted order ✓

D) Level order

Explanation: Inorder traversal sorted sequence دیتا ہے۔

Q2. Preorder traversal sequence؟

A) Left–Root–Right

B) Root–Left–Right ✓

C) Left–Right–Root

D) Right–Left–Root

Explanation: Root سے پہلے سب سے پہلے visit ہوتا ہے۔

Q3. Postorder traversal sequence؟

A) Root–Left–Right

B) Left–Root–Right

C) Right–Left–Root

D) Left–Right–Root ✓

Explanation: Root سے آخر میں visit ہوتا ہے۔

Q4. Tree traversal کا use؟

A) Insert

B) Delete

C) Visit nodes systematically ✓

D) Balance

Explanation: Traversal nodes کو specific order میں visit کرتا ہے۔

Q5. Preorder traversal کہاں useful ہے؟

A) Sorting

B) Tree copy ✓

C) Searching

D) Deleting

Explanation: Preorder structure copy میں use کرنے میں ہوتا ہے۔

□ Lab 6: Call by Value / Reference / Pointer

Q1. Call by value میں changes کہاں ہوتی ہیں؟

A) Original variables

B) Memory

C) Copies only ✓

D) Heap

Explanation: Original values affect نہیں ہوتیں۔

Q2. Call by reference میں parameter کیا pass ہوتا ہے؟

A) Copy

B) Address

C) Reference ✓

D) Value

Explanation: Original variable directly access ہوتا ہے۔

Q3. Pointer function میں data access کیسے ہوتا ہے؟

A) Value

B) Index

C) Reference

D) Address via pointer ✓

Explanation: Pointer address use کرتا ہے۔

Q4. Template functions کا فائدہ؟

A) Speed

B) Memory

C) Multiple data types support ✓

D) Security

Explanation: One function for all data types۔

Q5. Swap function effective کب ہوتا ہے؟

A) Call by value

B) Reference & Pointer ✓

C) Copy

D) Constructor

Explanation: Original variables modify ہوتے ہیں۔

□ Lab 7–9: AVL Tree

Q1. AVL tree کیا ہے؟

A) Normal BST

B) Unbalanced tree

C) Self-balancing BST ✓

D) Heap

Explanation: AVL tree height balanced ہوتی ہے۔

Q2. AVL balance factor range؟

A) 0 to 2

B) -1 to +1 ✓

C) -2 to +2

D) 1 to 3

Explanation: Balance factor = height(left) – height(right)۔

Q3. AVL imbalance fix کیسے ہوتا ہے؟

A) Sorting

B) Traversal

- C) Rotation ✓
- D) Deletion

Explanation: Rotations tree کو balance کرتی ہیں۔

Q4. LL rotation کب ہوتی ہے؟

- A) Right-heavy
- B) Left-left heavy ✓
- C) Left-right
- D) Right-left

Explanation: Left subtree میں left insertion پر LL case-

Q5. AVL tree search complexity؟

- A) $O(n)$
- B) $O(n^2)$
- C) $O(\log n)$ ✓
- D) $O(1)$

Explanation: Height balanced سے $\log n$ کی وجہ سے

□ Lab 10: Huffman Encoding

Q1. Huffman coding کس چیز کے لیے ہے؟

- A) Sorting
- B) Compression ✓
- C) Searching
- D) Encryption

Explanation: Data compression کے لیے use ہوتی ہے۔

Q2. Huffman tree پر بنتا ہے؟ basis کس

- A) Random
- B) Frequency ✓
- C) Index
- D) Size

Explanation: Characters کی frequency use ہوتی ہے۔

Q3. Huffman tree کیا produce کرتا ہے؟

- A) Fixed codes
- B) Same length codes
- C) Variable length codes ✓
- D) Binary search

Explanation: Frequent characters کو short codes میں ملتے ہیں۔

Q4. Huffman tree میں leaf nodes کیا represent کرتے ہیں؟

- A) Frequency
- B) Characters ✓

- C) Root
- D) Levels

Explanation: Leaf nodes characters ہوتے ہیں۔

Q5. Huffman coding greedy algorithm ہے؟

- A) No
- B) Sometimes
- C) Yes ✓
- D) Recursive

Explanation: Greedy approach use ہوتی ہے۔

□ Lab 11–12: Min Heap

Q1. Min heap میں root کیا store کرتا ہے؟

- A) Max value
- B) Min value ✓
- C) Random
- D) Last value

Explanation: Root ہمیشہ minimum value رکھتا ہے۔

Q2. Heap complete binary tree ہوتا ہے؟

- A) No
- B) Sometimes
- C) Yes ✓
- D) BST

Explanation: Heap always complete binary tree ہوتا ہے۔

Q3. Insert operation میں کیا use ہوتا ہے؟

- A) Heapify up ✓
- B) Heapify down
- C) Traversal
- D) Rotation

Explanation: Insert کے بعد upward adjustment ہوتی ہے۔

Q4. buildHeap method complexity؟

- A) $O(n)$ ✓
- B) $O(n \log n)$
- C) $O(\log n)$
- D) $O(1)$

Explanation: buildHeap linear time میں بنتا ہے۔

Q5. Heap کہاں use ہوتا ہے؟

- A) Sorting
- B) Priority Queue ✓

- C) Hashing
- D) Graph

Explanation: Priority queue implementation heap سے ہوتی ہے۔

□ Lab 13: Union-Find (Union by Size)

Q1. Union-Find data structure کس لیے ہے؟

- A) Sorting
- B) Searching
- C) Disjoint sets ✓
- D) Stack

Explanation: Different sets manage کرنے کے لیے۔

Q2. Union by size attach میں کیا ہوتا ہے؟

- A) Bigger tree to smaller
- B) Smaller tree to bigger ✓
- C) Random
- D) Root to leaf

Explanation: Tree height minimize کرنے کے لیے۔

Q3. Find operation کیا کرتا ہے؟

- A) Insert
- B) Delete
- C) Root find کرتا ہے ✓
- D) Sort

Explanation: Set representative find کرتا ہے۔

Q4. Union-Find complexity optimized کیسے ہوتی ہے؟

- A) Traversal
- B) Path compression ✓
- C) Rotation
- D) Hashing

Explanation: Path compression tree flatten کرتا ہے۔

Q5. Union-Find use کہاں ہوتا ہے؟

- A) DFS
- B) MST (Kruskal) ✓
- C) Stack
- D) Heap

Explanation: Graph algorithms میں use ہوتا ہے۔

□ Lab 14: Binary Search

Q1. Binary search کس condition پر possible ہے؟

- A) Unsorted array
- B) Sorted array ✓
- C) Random
- D) Linked list

Explanation: Binary search sorted data پر work کرتا ہے۔

Q2. Binary search complexity؟

- A) $O(n)$
- B) $O(n^2)$
- C) $O(\log n)$ ✓
- D) $O(1)$

Explanation: Every step half data eliminate کرتا ہے۔

Q3. Binary search technique؟

- A) Linear
- B) Divide and conquer ✓
- C) Greedy
- D) Brute force

Explanation: Divide and conquer strategy use ہوتی ہے۔

Q4. Middle element compare کس سے ہوتا ہے؟

- A) Last
- B) First
- C) Key ✓
- D) Index

Explanation: Target key سے comparison ہوتا ہے۔

Q5. Binary search recursive بھی ہو سکتی ہے؟

- A) No
- B) Sometimes
- C) Yes ✓
- D) Never

Explanation: Recursive and iterative دونوں versions ہوتے ہیں۔

□ Lab 15: Hashing (Linear Probing)

Q1. Hashing کیا ہے؟

- A) Sorting
- B) Mapping keys to index ✓
- C) Searching tree

D) Traversal

Explanation: Key کو index میں convert کیا جاتا ہے۔

Q2. Collision کب ہوتا ہے؟

A) Same key

B) Same index for different keys ✓

C) Table empty

D) Hash full

Explanation: Two keys same index پر map ہوں۔

Q3. Linear probing کیا ہے؟

A) Random search

B) Backtracking

C) Sequential empty slot search ✓

D) Tree traversal

Explanation: Next free index search کیا جاتا ہے۔

Q4. Linear probing drawback؟

A) No collision

B) Clustering ✓

C) Fast access

D) Simple

Explanation: Primary clustering issue ہوتا ہے۔

Q5. Hash table average search complexity؟

A) $O(n)$

B) $O(\log n)$

C) $O(1)$ ✓

D) $O(n^2)$

Explanation: Ideal case constant time۔

□ Lab 16: Sorting Algorithms

Q1. Bubble sort complexity (worst case)؟

A) $O(n)$

B) $O(\log n)$

C) $O(n^2)$ ✓

D) $O(1)$

Explanation: n^2 کی وجہ سے Nested loops

Q2. Selection sort swaps کتنی بار کرتا ہے؟

A) n^2

B) n

C) $n-1$ ✓

D) $\log n$

Explanation: swap میں ایک pass ہر

Q3. Insertion sort best case؟

A) $O(n^2)$

B) $O(\log n)$

C) $O(n)$ ✓

D) $O(1)$

Explanation: Already sorted array میں linear time-

Q4. Elementary sorting algorithms کون سے ہیں؟

A) Merge, Quick

B) Heap, Radix

C) Bubble, Selection, Insertion ✓

D) DFS

Explanation: Basic sorting algorithms-

Q5. Stable sort کون سا ہے؟

A) Selection

B) Bubble ✓

C) Heap

D) Quick

Explanation: Bubble sort relative order maintain کرتا ہے۔