Data Structures — CS214

Midterm Practice Pack

Models: 471 | 462 Alt | 462 DS | 461 | 452 | 451

Contents

1) Repeated Topics — Focus Here First
2) Model 471
3) Model 462 (Alternative)
4) Model 462 (Data Structures)
5) Model 461
6) Model 452
7) Model 451

1) Repeated Topics — Study These First

These topics appear across ALL models. Mastering them covers most of your exam points.

1. Linear Data Structures
Stores data sequentially; each element linked to previous and next. Examples: Array, Stack, Queue, Linked List.

2. Static vs Dynamic
Static = fixed size (Array). Dynamic = grows/shrinks at runtime (Linked List, Tree). Based on memory allocation.

3. Primitive vs Non-Primitive
Primitive = single value, one type (int, char). Non-Primitive = built from primitives (Array, LL, Stack, Tree).

4. Contiguous Memory (Array)
Array = contiguous memory. Linked List = NOT contiguous (connected via pointers).

5. Traversal
Accessing every element in a data structure exactly once.

6. FIFO / Queue & LIFO / Stack
Queue = First In First Out (insert at rear, delete at front). Stack = Last In First Out (top = most recently inserted).

7. Memory Address Formulas
Row-major: Address = Base + W × (i × N + j)
Column-major: Address = Base + W × (j × M + i)
M = rows, N = cols, W = word size, i = row index, j = col index

8. Linked List vs Array
LL: dynamic size, easy insert/delete. Array: random access, less memory per element.

9. Insertion in Linked List
Can insert at beginning, end, or middle. Middle insertion = modify exactly 2 pointers.

10. Code Patterns
Array: for loop + condition + modify/print.
Linked List: Node* temp=head; while(temp!=NULL){ ... ; temp=temp->link; }

2) Model 471

Part I: Multiple Choice [12 points]

1) To build a Linked List, where can you insert a new node?
A. At the beginning
B. At the end
C. In the middle
D. All of the above

2) Which of the following are Linear Data Structures?
A. Array
B. Stack
C. Linked List
D. All of the above

3) Which factor(s) influence the choice of a specific Data Structure?
A. Complexity of implementation
B. Efficiency of required operations
C. Amount of memory space needed
D. All of the above

4) Which structure is preferred when size frequently changes and memory should be flexible?
A. Array
B. Linked List
C. Hash Table
D. Tree

5) Which of the following is a Static Data Structure?
A. Linked List
B. Stack
C. Array
D. Queue

6) Which of the following is a logical representation of data?
A. Array
B. Linked List
C. Stack
D. None of these

7) A structure used to store only a single value of one data type is:
A. Non-Primitive Data Structure
B. Primitive Data Structure
C. Abstract Data Type
D. Dynamic Data Structure

8) What is a key difference between static and dynamic data structures?
A. Ease of implementation
B. Size flexibility
C. Time complexity of traversal
D. Whether linear or non-linear

9) Operation to combine two sorted arrays into one sorted structure?
A. Traverse
B. Separate
C. Merge
D. Split

10) Most basic data structure for direct indexing by position?
A. Stack
B. Linked List
C. Queue
D. Array

11) Examining every element in a data structure exactly once is:
A. Searching
B. Sorting
C. Traversal
D. Allocation

12) Which structure implements FIFO?
A. Stack
B. Array
C. Queue
D. Tree

Answer Key — 471 — MCQ

✓ **1) D — All of the above**
✓ **2) D — All of the above**
✓ **3) D — All of the above**
✓ **4) B — Linked List**
✓ **5) C — Array**
✓ **6) C — Stack**
✓ **7) B — Primitive Data Structure**
✓ **8) B — Size flexibility**
✓ **9) C — Merge**
✓ **10) D — Array**
✓ **11) C — Traversal**
✓ **12) C — Queue**

Part II: True or False [4 points]

[ ] 1) An ADT is concerned with implementation details rather than the logical structure.
[ ] 2) An Array is a Physical data structure because its elements are stored in contiguous memory locations.
[ ] 3) Non-primitive data structures are generally easier to implement than linear data structures.
[ ] 4) A Linked List is a Non-Primitive, Linear data structure, and a stack data structure.

Answer Key — 471 — T/F

✓ **1) FALSE** — ADT defines WHAT a structure does, not HOW.
✓ **2) TRUE** — Adjacent memory = physical structure.
✓ **3) FALSE** — Non-primitive structures are MORE complex.
✓ **4) FALSE** — LL is not a Stack.

Part III: Written

Q3: Memory Address — Column Major. Base=100, W=2. int mat[2][4]={{1,2,3,4},{5,6,7,8}}. Find address of element 7.

Answer:

```
Element 7 at [1][2] -> i=1, j=2, M=2
LOC = 100 + 2 × (2×2 + 1) = 100 + 10 = 110
```

Q4: Reverse arr[5]={40,18,7,3,5} then print elements < 10.

Answer:

```
  void reverseAndPrint() {
    int arr[5]={40,18,7,3,5};
    for(int i=0;i<5/2;i++){
      int t=arr[i];
      arr[i]=arr[4-i];
      arr[4-i]=t;
    }
    for(int i=0;i<5;i++)
      if(arr[i]<10) cout<<arr[i]<<" ";
  }
  // After reverse: {5,3,7,18,40} -> prints: 5 3 7
```

3) Model 462 (Alternative)

MCQ [5 points]

1) The main property of a Circular Queue is that it:
A. Writes more data even after reaching maximum size
B. Overwrites data directly on top of old data
C. (other option)
D. Wraps around the array to fill empty positions

2) In a Circular Queue, which condition means the queue is full?
A. (other condition)
B. Front is ahead of rear by one
C. Rear is ahead of front by one
D. Front and rear are equal

3) The top in a stack represents:
A. First element
B. Last element
C. Index 1
D. None

4) Which is correct about a priority queue?
A. First-added always processes first regardless of priority
B. If same priority, first-added processes first
C. Lower priority processed before higher priority
D. Priority does not affect order

5) Disadvantage of an array data structure:
A. Array address is the first element address
B. Accessing data randomly
C. (other option)
D. Wasted memory if fewer elements than allocated size

Answer Key — 462 Alt — MCQ

✓ **1) D — Wraps around the array**
✓ **2) C — Rear is ahead of front by one**
✓ **3) B — Last element**
✓ **4) B — Same priority: first-added first**
✓ **5) D — Wasted memory**

True or False [5 points]

[ ] 1) Logical data structure can be created independently.
[ ] 2) We can access the middle element in a stack directly.
[ ] 3) A queue adds an element on one end and deletes from the other end.
[ ] 4) Traversal accesses each element of a data structure once.
[ ] 5) Execution time is a factor for choosing a data structure.

Answer Key — 462 Alt — T/F

✓ **1) TRUE** — ADT defines behavior independently of implementation.
✓ **2) FALSE** — Stack only exposes the top.
✓ **3) TRUE** — Enqueue at rear, dequeue at front.
✓ **4) TRUE** — Definition of traversal.
✓ **5) TRUE** — Time complexity is a key DS selection factor.

Written

W1: Remove even elements from an array of 20 integers and display remaining.

Answer:

```
void removeEven() {
  int arr[20]={...}; int newArr[20]; int j=0;
  for(int i=0;i<20;i++)
    if(arr[i]%2!=0) newArr[j++]=arr[i];
  for(int i=0;i<j;i++)
    cout<<newArr[i]<<" ";
}
```

W2: Enumerate the linked list. Head -> [14] -> [30] -> [25] -> [42] -> NULL

Answer:

```
Node* temp=head;
while(temp!=NULL){
  cout<<temp->data<<" ";
  temp=temp->link;
}
// Output: 14 30 25 42
```

4) Model 462 (Data Structures)

Multiple Choice [10 points]

1) Linear data structures arrange data:
A. Independently
B. Hierarchically
C. Sequentially
D. Non-consecutively

2) Elements of _____ are stored in contiguous memory locations.
A. Array
B. Linked list
C. Queue
D. Tree

3) Given: head->90->52->45, the value of head->link->data is:
A. 10
B. 90
C. 52
D. 45

4) Primary advantage of linked list over array?
A. Faster element access
B. Dynamic size and ease of insertion/deletion
C. Fixed size
D. Better memory utilization

5) Static DS is more suitable when:
A. Size is unknown and changes frequently
B. Memory must be minimized
C. DS stored in external memory
D. Data needs to be accessed randomly and frequently

6) Result of deleting 2nd element from arr={10,17,3,12,25}?
A. {10,3,12,25}
B. {10,3,12,25,0}
C. {10,0,3,12,25}
D. {10,3,12,25,17}

7) Which data structure is LIFO?
A. Stack
B. Queue
C. Array
D. Tree

8) If row-major array uses column-major formula for retrieval, what happens?
A. Always accesses first element
B. Always accesses last element of first row
C. An element from a different row is retrieved
D. None

9) Which property is associated with a queue?
A. First in Last Out
B. First in First Out
C. Last in First Out
D. All of the above

10) Which is NOT a dynamic data structure?
A. Tree
B. Linked list
C. Stack
D. None of the above (all are dynamic)

Answer Key — 462 DS — MCQ

✓ **1) C — Sequentially**
✓ **2) A — Array**
✓ **3) C — 52**
✓ **4) B — Dynamic size + easy insert/delete**
✓ **5) D — Random access needed**
✓ **6) B — {10,3,12,25,0}**
✓ **7) A — Stack**
✓ **8) C — Element from different row**
✓ **9) B — First in First Out**
✓ **10) D — None (all are dynamic)**

True or False [5 points]

[ ] 1) Classification of data structures based on memory allocation is static and dynamic.
[ ] 2) Elements of linked lists are stored in adjacent memory locations.
[ ] 3) Linked lists require more memory per element than arrays.
[ ] 4) Column-major and row-major give identical memory addresses for symmetric matrices.
[ ] 5) Insertion of an element in an array is easier than in a linked list.

Answer Key — 462 DS — T/F

✓ **1) TRUE** — Static (Array) vs Dynamic (LL).
✓ **2) FALSE** — LL nodes connect via pointers, not adjacency.
✓ **3) TRUE** — Each node holds data + pointer.
✓ **4) FALSE** — The two formulas always compute different addresses.
✓ **5) FALSE** — Array insertion requires shifting; LL just updates 2 pointers.

Written

W1: Write function copyMatrix — copies a 2x4 matrix.

Answer:

```
void matrix::copyMatrix(matrix &m1) {
  for(int i=0;i<2;i++)
    for(int j=0;j<4;j++)
      Matrix[i][j]=m1.matrix[i][j];
}
```

W2: Write function countOddNumbers for a linked list.

Answer:

```
int countOddNumbers() {
  int count=0; Node* temp=p;
  while(temp!=NULL){
    if(temp->data%2==1) count++;
```

```
    temp=temp->link;
  }
  return count;
}
```

5) Model 461

Question 1: Multiple Choice [10 points]

1) In a linked list, each node contains a data field and a second field that is a:
A. Pointer refers to character
B. Pointer refers to node
C. Pointer refers to integer
D. Node

2) _____ uses computer memory effectively.
A. Array
B. Linked list
C. Queue
D. Tree

3) Inserting at the middle of a singly linked list requires modifying how many pointers?
A. 1
B. 2
C. 3
D. 4

4) When head == NULL in a linked list, the list is:
A. empty
B. with one node
C. full
D. overflow

5) Which is a characteristic of a primitive data structure?
A. Stores values of a single data type
B. Can be modified dynamically
C. Utilizes multiple memory levels
D. Difficult to implement

6) Given A[3]={5,7,4}, after insert(1,2), what is the result?
A. 5,7,4,1,2
B. 1,2,5,7,4
C. 5,7,1,4
D. 5,2,7,4

7) In _____ data structure, every element is attached to its previous and next adjacent.
A. dynamic
B. logical
C. linear
D. non-linear

8) Which data structure is used to perform searches frequently?
A. Stack
B. Queue
C. Array
D. Tree

9) Which data structure is used for efficient indexing?
A. Array
B. Linked list
C. Queue
D. Stack

10) Which data structure is suitable for frequent insertions and deletions?
A. Array
B. Linked list
C. Stack
D. Tree

Answer Key — 461 — MCQ

✓ **1) B — Pointer refers to node**
✓ **2) B — Linked list**
✓ **3) B — 2 pointers**
✓ **4) A — empty**
✓ **5) A — Single data type**
✓ **6) D — 5,2,7,4 (insert value 2 at position 1)**
✓ **7) C — linear**
✓ **8) C — Array**
✓ **9) A — Array**
✓ **10) B — Linked list**

Question 2: True or False [5 points]

[ ] 1) Traversal is an operation used to access each element of a data structure once.
[ ] 2) To access the fifth element in array A, we use A[5].
[ ] 3) Linear data structures are arranged in a hierarchical manner.
[ ] 4) To compare two linked lists we can use a recursive function.
[ ] 5) The elements of linked lists are stored in adjacent memory locations.

Answer Key — 461 — T/F

✓ **1) TRUE** — Standard definition.
✓ **2) FALSE** — Arrays are 0-indexed; 5th = A[4].
✓ **3) FALSE** — Linear = sequential, not hierarchical.
✓ **4) TRUE** — Recursive comparison works node by node.
✓ **5) FALSE** — LL nodes are linked by pointers, not adjacent.

Question 3: Memory Address — Row Major [2 points]

Find address of element 9. Base=110, W=4. int mat[3][3]={{3,6,8},{5,4,1},{2,9,7}}.

Answer:

```
Element 9 at [2][1] -> i=2, j=1, N=3
LOC = 110 + 4 × (2×3 + 1) = 110 + 28 = 138
```

Question 4: Delete duplicated elements. Array size = MAX [3 points]

Answer:

```
void del_dup() {
  for(int i=0;i<MAX;i++)
    for(int j=i+1;j<MAX;j++)
      if(arr[i]==arr[j]){
        for(int k=j;k<MAX-1;k++) arr[k]=arr[k+1];
```

```
        arr[MAX-1]=0;
        j--;   // re-check new element at position j
      }
}
```

6) Model 452

Instructor: Dr. Khalil Al-Shuqairat | Term 452 | Coverage: First 2 Chapters

Question 1: Multiple Choice [5 points]

1) Which of the following is an example of a data structure?
A. Variable
B. Array
C. Function
D. Loop

2) Which of the following is a characteristic of a static data structure?
A. Fixed size in memory
B. Can store a variable amount of data
C. Utilizes hierarchical multiple levels
D. None of the above

3) Which of the following is a characteristic of a linear data structure?
A. Elements arranged independently
B. Elements arranged sequentially
C. Complex implementation
D. Inefficient memory utilization

4) Which data structure is suitable for frequent insertions and deletions?
A. Array
B. Linked list
C. Binary search tree
D. Stack

5) What is the main purpose of a data structure?
A. To increase memory usage
B. To slow down data processing
C. To organize data efficiently
D. All of the above

Answer Key — 452 — MCQ

✓ **1) B — Array**
✓ **2) A — Fixed size in memory**
✓ **3) B — Elements arranged sequentially**
✓ **4) B — Linked list**
✓ **5) C — Organize data efficiently**

Question 2: True or False [3 points]

[ ] 1) An array is an example of a primitive data structure.
[ ] 2) An ADT defines the behavior of objects through a set of values and operations.
[ ] 3) The size of dynamic data structures can be changed.

Answer Key — 452 — T/F

✓ **1) FALSE** — Array is non-primitive.
✓ **2) TRUE** — ADT defines what operations exist.
✓ **3) TRUE** — Allocated/freed at runtime.

Question 3: Memory Address — Column Major [3 points]

Find address of element 2. Base=105, W=4. int mat[2][3]={{3,6,8},{5,4,2}}.

Answer:

```
Element 2 at [1][2] -> i=1, j=2, M=2
LOC = 105 + 4 × (2×2 + 1) = 105 + 20 = 125
```

Question 4: Add Two 3x3 Matrices [3 points]

Answer:

```
void matrix::matadd(matrix &m1, matrix &m2) {
  int mat[3][3];
  for(int i=0;i<3;i++)
    for(int j=0;j<3;j++)
      mat[i][j] = m1.mat[i][j]
                + m2.mat[i][j];
}
```

Question 5: Display 4x4 Matrix [3 points]

Answer:

```
void matrix::display() {
  for(int i=0;i<4;i++){
    for(int j=0;j<4;j++)
      cout<<mat[i][j]<<" ";
    cout<<endl;
  }
}
```

Question 6: Delete Duplicated Elements. Array size = 10 [3 points]

Answer:

```
void array::rem_dup() {
  for(int i=0;i<10;i++)
```

```
    for(int j=i+1;j<10;j++)
      if(a[i]==a[j]){
        for(int k=j;k<10-1;k++) a[k]=a[k+1];
        j--;   // re-check new element
      }
}
```

7) Model 451

Qassim University | CS214 Data Structures | Third Semester 1444/1445 | Duration: 50 min

Question 1: Multiple Choice [5 points]

1) Data structure is:
A. A systematic way to organize data so it can be used efficiently
B. A list of a finite number (n) of similar data
C. A linear collection of data elements, called nodes
D. All of the above

2) How can we describe an array in the best possible way?
A. The Array shows a hierarchical structure
B. Arrays are unchangeable
C. The Array is not a data structure
D. Container that stores elements of similar types

3) Which of the following is a logical data structure?
A. Array
B. Linked list
C. Stack
D. None of the above

4) A pointer is:
A. A linear collection of data elements, called nodes
B. A variable which contains the address in memory of another variable
C. A systematic way to organize data so it can be used efficiently
D. None of the above

5) Which of the following is a linear data structure?
A. Stack
B. Queue
C. Linked list
D. All of the above

Answer Key — 451 — MCQ

✓ **1) A — Systematic way to organize data**
✓ **2) D — Container of similar types**
✓ **3) C — Stack**
✓ **4) B — Variable holding a memory address**
✓ **5) D — All of the above**

Question 2: True or False [3 points]

[ ] 1) The Array is not a suitable data structure to store much data.
[ ] 2) In data structures, the Sort operation arranges all data items in order.
[ ] 3) Execution time is a factor in choosing an appropriate data structure.

Answer Key — 451 — T/F

✓ **1) FALSE** — Array IS suitable for large data.
✓ **2) TRUE** — Sort reorders elements.
✓ **3) TRUE** — Time complexity is a main factor.

Question 3: Memory Address — Column Major [2 points]

Find address of element 7. Base=100, W=2. 2x4 array: Row 0={1,2,3,4}, Row 1={5,6,7,8}.

Answer:

```
Element 7 at [1][2] -> i=1, j=2, M=2
LOC = 100 + 2 × (2×2 + 1) = 100 + 10 = 110
```

Question 4: Add Two 3x3 Matrices [2 points]

Answer:

```
void matrix::matadd(matrix &m1, matrix &m2) {
  int mat[3][3];
  for(int i=0;i<3;i++)
    for(int j=0;j<3;j++)
      mat[i][j] = m1.mat[i][j]
               + m2.mat[i][j];
}
```

Question 5: Display Array of Size 8 [2 points]

Answer:

```
void display() {
  for(int i=0;i<8;i++)
    cout<<arr[i]<<" ";
  cout<<endl;
}
```

Question 6: Delete One Element from array of size 5 at position Pos [3 points]

Answer:

```
void del(int Pos) {
  int i;
```

```
    for(i=Pos; i<5; i++) arr[i-1]=arr[i];
    arr[4]=0;
  }
```

Question 7: Transpose 4x4 Matrix [3 points]

Answer:

```
  void matrix::transpose(matrix &m1) {
    int mat[4][4];
    for(int i=0;i<4;i++)
      for(int j=0;j<4;j++)
        mat[i][j]=m1.mat[j][i];
  }
```

Good Luck! Focus on the repeated topics in Section 1 — those are your guaranteed marks.