I've reviewed the Algorithms and Complexity exam papers from Royal Holloway, University of London spanning from 2017 to 2024. Let me make notes on the common themes and recurring question topics across these exams.

**Common Themes and Question Types**

**1. Order of Growth Analysis**

* Almost every exam begins with questions asking to express functions in Θ notation and sort them by growth rate
* Often includes follow-up questions about Big-O notation definitions, constants, and bounds
* Typically worth around 10-12 marks

**2. Algorithm Time Complexity Analysis**

* Analysis of running time for given code snippets/functions
* Requires justification of Big-O notation results
* Usually includes recursive functions or nested loops
* Worth 10-12 marks on average

**3. Data Structures**

* **Binary Search Trees (BST)**:
  + Operations (insertion, deletion)
  + AVL trees and balancing properties
  + Rotations to maintain balance
* **Heaps**:
  + Min-heap properties
  + Operations (makeHeap, insertion, deleteMin)
  + Tree representation of arrays
* **Linked Lists**:
  + Structure and operations
  + Time complexity of operations
* **Hash Tables**:
  + Open addressing
  + Linear probing
  + Hash function applications

**4. Sorting Algorithms**

* Merge sort
* Quick sort
* Insertion sort
* Analysis of time complexity (best/worst/average cases)
* Illustrating algorithm execution on arrays

**5. Graph Algorithms**

* **Dijkstra's Algorithm**:
  + Finding shortest paths
  + Implementation efficiency
  + Shortest path trees
* **Depth-First Search (DFS) and Breadth-First Search (BFS)**
* **Minimum Spanning Trees**:
  + Kruskal's algorithm
  + Prim's algorithm
* **Connected Components**

**6. Algorithm Design for Real-world Problems**

* Common practical problems include:
  + Finding artists with most songs in a collection
  + Event scheduling systems
  + Travel/route planning
  + Email management systems
  + Plagiarism detection

**7. Recursive Algorithms**

* Recursion trees
* Function output calculation
* Time complexity analysis
* Memoization technique application

**8. Divide and Conquer**

* Understanding the strategy
* Application to specific problems (e.g., mergesort)
* Recurrence relations

**Question Structure Patterns**

1. Most exams have 7-8 questions total
2. Two-hour time limit is consistent across all exams
3. Questions generally range from 6-16 marks each
4. Many questions have multiple parts that build upon each other
5. Questions often ask to both explain algorithms and demonstrate them on examples
6. Practical application questions frequently appear at the end of the exam

**Mark Distribution**

The exams appear to be out of approximately 100 marks total, with:

* Theoretical questions (notations, definitions): ~20-25%
* Algorithm analysis: ~20-25%
* Data structure operations: ~25-30%
* Graph algorithms: ~15-20%
* Practical application problems: ~15-20%

These patterns suggest students should focus on understanding the core theoretical concepts, practicing algorithm analysis, mastering common data structures and operations, and applying these to practical problems.