



**University of Colombo School of Computing**  
**SCS 1308 - Foundations of Algorithms**  
**Take-Home 08**

---

**Question 1: Exploring Searching Algorithms**

- (a) Explain the purpose and importance of searching algorithms in computer science. Discuss the characteristics and use cases of linear search, binary search, jump search, and interpolation search.
- (b) Compare the time complexity of linear, binary, jump, and interpolation search. Provide examples of practical scenarios where each algorithm would be most effective.
- (c) Design pseudocode for both jump search and interpolation search. Apply each algorithm to a sample dataset [10, 20, 30, 40, 50, 60, 70, 80, 90] to locate the value 70. Illustrate the steps involved.
- (d) Critically analyze the trade-offs between time complexity, memory usage, and dataset properties (e.g., sorted vs. unsorted, uniformly distributed) when selecting a search algorithm. Relate your analysis to real-world applications like web search engines and AI pathfinding.

**Question 2: Understanding Sorting Algorithms**

- (a) Discuss the concept of stability in sorting algorithms. Provide examples of stable and unstable sorting algorithms and explain their relevance in real-world scenarios such as database management and genetic data analysis.
- (b) Compare shell sort and radix sort in terms of time complexity, memory usage, and practical applications. For each algorithm, identify scenarios where it would be more advantageous than others.
- (c) Develop pseudocode for shell sort and radix sort. Apply shell sort to [45, 23, 11, 89, 77, 98, 4, 28, 65, 43] and radix sort to [170, 45, 75, 90, 802, 24, 2, 66]. Illustrate the sorting process with each step.
- (d) Critically evaluate how sorting algorithms like shell sort and radix sort handle large datasets. Discuss the importance of stability, memory overhead, and special conditions like fixed-length keys or partially ordered data. Include recommendations for selecting appropriate algorithms in specific applications such as graphics processing and data indexing.