PYQ of (Analysis and Design of Algorithms) 1 Mark

2019

a) Write difference between θ notation and O notation.
b) Write the worst care time complexity of insertion sort and heap sort.
c) What is meant by amortized analysis?
d) For what type of problems dynamic programming algorithms are useful?
e) What is meant by Greedy algorithm?
f) Define Red-Black tree.
g) What is the basic difference between Red-Black tree and Binary Search tree?
h) What is the basic principle of counting sort?

2021

a) What is algorithm?
b) Why do we use algorithm?
c) What is big O notation?
d) Name an iterative searching technique.
e) What is the time complexity of quick sort in worst case?
f) What is the time complexity of binary search in worst case?
g) What is graph?
h) What do you mean by spanning tree?

2022

a. Symbolic analysis of run time is advantageous over estimating the actual run time – Justify.
b. Why big-Omega (Ώ) notation is used?
c. Name an algorithm design paradigm that necessarily involves heuristics?
d. Name a sorting algorithm which yields the same time complexities both in average case and worst case.
e. State a real life example of decision tree.
f. Define tree as a graph.
g. Name an inventor of AVL tree.
h. What do you mean by spanning tree?

2022-23

a) When do you use empirical approach for determining computational complexity?
b) State whether time complexity is platform dependent or not.
c) Which design paradigm do we use in quick sort algorithm?
d) Name an algorithm that employs Geedy paradigm.
e) Name a sorting algorithm which yields the same time complexity both in average case and worst case.
f) Name a linear time sorting technique.
g) What is AVL tree?
h) Why do we use KMP technique?

ANSWERS

2019

a) Write difference between θ notation and O notation.

→ **Θ (Theta)** gives a tight (exact) bound, while **O (Big-O)** gives only an upper bound (worst-case).

b) Write the worst-case time complexity of insertion sort and heap sort.

→ Insertion Sort: **O(n²)**

→ Heap Sort: **O(n log n)**

c) What is meant by amortized analysis?

→ It gives average time per operation over a sequence, even if individual operations are expensive.

d) For what type of problems are dynamic programming algorithms useful?

→ For problems with *overlapping subproblems* and *optimal substructure*.

e) What is meant by Greedy algorithm?

→ A greedy algorithm makes the best local choice at each step to reach a global optimum.

f) Define Red-Black tree.

→ A self-balancing binary search tree where each node has a color (red or black) and follows certain rules to maintain balance.

g) What is the basic difference between Red-Black tree and Binary Search Tree?

→ Red-Black tree is always balanced; Binary Search Tree may become unbalanced.

h) What is the basic principle of counting sort?

→ It counts the frequency of elements and uses this to place elements directly in sorted order.

2021

a) What is algorithm?

→ A set of steps to solve a problem.

b) Why do we use algorithm?

→ To solve problems in an efficient and logical way.

c) What is big O notation?

→ It shows the upper bound of time complexity (worst-case performance).

d) Name an iterative searching technique.

→ **Binary Search**

e) What is the time complexity of quick sort in worst case?

→ **O(n²)**

f) What is the time complexity of binary search in worst case?

→ **O(log n)**

g) What is graph?

→ A set of vertices (nodes) and edges connecting them.

h) What do you mean by spanning tree?

→ A subgraph that connects all nodes of a graph without any cycles.

2022

a) Symbolic analysis of run time is advantageous over estimating the actual run time – Justify.

→ Symbolic analysis gives a general performance idea, independent of hardware or input size.

b) Why big-Omega (Ω) notation is used?

→ To represent the best-case lower bound of an algorithm.

c) Name an algorithm design paradigm that necessarily involves heuristics.

→ **Greedy paradigm**

d) Name a sorting algorithm which yields the same time complexities both in average case and worst case.

→ **Merge Sort** – O(n log n) in both cases.

e) State a real-life example of decision tree.

→ *Medical diagnosis* or *loan approval system*.

f) Define tree as a graph.

→ A tree is a connected acyclic graph.

g) Name an inventor of AVL tree.

→ **Adelson-Velsky and Landis**

h) What do you mean by spanning tree?

→ A spanning tree connects all vertices of a graph with minimum number of edges and no cycles.

2022–23

a) When do you use empirical approach for determining computational complexity?

→ When theoretical analysis is hard or unclear.

b) State whether time complexity is platform dependent or not.

→ No, it is platform-independent.

c) Which design paradigm do we use in quick sort algorithm?

→ **Divide and Conquer**

d) Name an algorithm that employs Greedy paradigm.

→ **Prim's Algorithm**

e) Name a sorting algorithm which yields the same time complexity both in average case and worst case.

→ **Merge Sort**

f) Name a linear time sorting technique.

→ **Counting Sort**

g) What is AVL tree?

→ A self-balancing binary search tree (BST).

h) Why do we use KMP technique?

→ For efficient pattern matching in strings.

INTERNAL ASSESSMENT 2023-24 Q&A

1. What is an algorithm? What is the need for an algorithm?

- **Definition:** An algorithm is a finite, step-by-step procedure or formula for solving a problem or performing a task.
- **Need:** Algorithms are essential for:
  - Providing clear instructions that can be followed to achieve a specific outcome.
  - Ensuring efficiency and effectiveness in problem-solving.

2. Write the characteristics of Divide and Conquer algorithm.

- **Characteristics:**
  - **Divide:** Break the problem into smaller subproblems.
  - **Conquer:** Solve the subproblems recursively.
  - **Combine:** Merge the solutions of the subproblems to get the final solution.
  - **Efficiency:** Often leads to faster run times compared to naive algorithms through better resource usage.

3. Why is quick sort called partition-exchange sort?

- **Partitioning:** Quick sort divides the array into smaller segments (partitions) based on a chosen pivot.
- **Exchange:** The elements are rearranged (exchanged) such that all elements less than the pivot come before it and all greater elements come after it, facilitating a more efficient sorting process.

4. What is an in-place sorting algorithm?

- **Definition:** An in-place sorting algorithm is one that requires a constant amount (O(1)) of additional space for sorting, meaning it sorts the list without needing extra storage for another array.
- **Example:** Quick sort and bubble sort are examples of in-place sorting algorithms.

5. What are the two phases of heap sort?

- **Phases:**
  - **Building the Heap:** Convert the array into a binary heap structure, which can be done in O(n) time.
  - **Sorting:** Repeatedly extract the maximum element from the heap and reconstruct the heap, which is done in O(n log n) time.

6. Why is heap sort better than selection sort?

- **Efficiency:** Heap sort has a guaranteed time complexity of O(n log n) in all cases, whereas selection sort operates at $O(n^2)$.
- **Data Structure:** Heap sort utilizes a binary heap, which allows for efficient maximum retrieval, making it generally faster for large datasets.

7. What are the time and space complexity in algorithms?

- **Time Complexity:** A measure that indicates the amount of time an algorithm takes to complete as a function of the input size, often expressed using Big O notation (e.g., O(n), O(log n)).
- **Space Complexity:** The total amount of memory space required by the algorithm to execute, including both the space for input values and additional space for variables, recursive calls, etc.

8. Write the worst-case time complexity of quick sort and merge sort.

- **Quick Sort:** The worst-case time complexity is $O(n^2)$, which occurs when the smallest or largest element is consistently chosen as the pivot.
- **Merge Sort:** The worst-case time complexity is O(n log n), as it consistently divides the data and then merges it back together.