Rules to Answer

Answer supported by evidence, and reasoning links answer and evidence.
Response is specific and detailed.
Presence of keywords and scientific terminology.

Rules to Answer

Refer content if unable to recall after 15-30sec of trying, then again retrieve from memory without help.

Primitive Data Types

Free Recall

Primitive Data Types

Data types that are defined by the sytem.
e.g. int, float, double, bool

User defined data types

Example

User defined data types

Most programming languages allow the users to define their own datatypes. e.g. struct in C

Data Structures

Explain

Data structure

A particular way of storing and orgainizing data in a computer so that it can be used efficiently

Abstract Data Types (ADTs)

Free Recall

Abstract Data Types

are of two parts
1. Declaration of data
2.Declaration of operation

Asymptotic Notation

Elaborate

Asymptotic Notation or Algorithm's growth rate

are languages that allow us to analyze an algorithm's running time by identifying its behavior as the input size for the algorithm increases.

Big-O (O) vs Big-Omega (Ω)

Main idea

Big-O

Asymptotic Notation for the worst case, or ceiling of growth for a given function.
Analogy: At most

Big-O

Provides us with **an asymptotic upper bound** for the growth rate of runtime of an algorithm

Big-O

Say f(n) is your algorithm runtime, and g(n) is an arbitary time complexity you are trying ot relate to your algorithm

Big-O

f(n) is O(g(n)), if for some constants c $(c>0)$ and n_0, **f(n)<=c*g(n)** for every input size $n(n>n_0$

Big-O Examples

Big-O Examples

$f(n) = 3\log n + 100$
$g(n) = \log n$

Big-Omega

Best case
Floor growth rate
Asymptotic lower bound
Analogy: At least

Big O and Big-Omega

Give the mathematical defination

Small-o and Small-omega (ω)

Elaborate

Small-o and Small-omega

Both are not asymptotically tight (no equal to sign)

Theta Θ

Explain

Theta

Asymptotical tight bound

$c1*g(n) < f(n) < c2*g(n)$

Recurrence

Main idea

Recurrence

An equation that recursively defines a sequence once one or more initial terms are given: each further term of the sequence or array is defined as a function of the preceding terms.

Recursion vs Recurrence

Explain

Recursion vs Recurrence

A recurrence relation uses recursion to create a sequence. Recursion is not limited to generation of sequences.

Recursion

Recursion is the repeated use of a procedure or action

Recurrence

In fact, a recurrence relation uses recursion to define a sequence.

Recurrence

This sequence is built in such a way that each term is defined as a combination of previous terms. The generation of such a sequence is a requirement in the definition

Recurrence

Recurrence is analyzed by telescoping

Binary Search

Main Idea

Binary Search

Search algorithm that finds the position of a target value within a sorted array

Binary Search

Elements are already sorted in ascending order
Divide and Conquer algorithm is used

Binary Search

Element x is either in the left half of the array or in the right half or not there at all

Binary Search

Compare x to middle element k, x > k (x not in left half), x < k (x not in right half)

Running time of Binary Search

Explain

Running time of Binary Search

$\lg_2 N$
binary search runs in O(lgN)

One word answer

In computer science you use $\lg_2 N$ or $\lg_{10} N$

In computer science you use $\lg_2 N$ or $\lg_{10} N$

$\lg_2 N$

Merging

Elaborate

Merging

Merging is not a divide and conquer algorithm, but part of mergesort algorithm

Merge Sort

Merge Sort is a divide and conquer algorithm

Merging

Take two sorted arrays of numbers and make a single array which is sorted of all of those numbers

Merge Sort

Type of Algorithm

Merge Sort

Explain the complete algorithm

Mergesort

First divide the list into smallest unit (1 element), then compare each element with the adjacent list to sort and merge two adjacent lists. Finally all the elements are sorted and merged.

Run time for mergesort

Run time for mergesort

O(n*lgn)

Insertion sort vs Mergesort

Which one is better and why?

Insertion sort vs Mergesort

Insertion sort is $O(n^2)$.
Mergesort is O(n*logn).
So, merge-sort is a superior algorithm in terms of its running time on large datasets.

Algorithm runtime

Insertion sort and Bubble sort

Insertion and bubble sort Runtime

Both has
Best: $\Omega(n)$
Worst: O(n^2)