What is Big O notation?

What is data structure? Explain Abstract Data Type (ADT) with an example.

Explain time and space complexity of algorithm.

Explain time and space complexity of algorithm.

What are linear and non-linear data structures?

Elaborate the statement: "Data Structure play important role in writing efficient and effective algorithm." [4]

Define Big-Oh and Theta notation with suitable example. [4]

Explain the importance of data structures and point out the areas in which data structures are being applied extensively. Explain why List is called as Abstract Data type (ADT). [2+2]

How do you find complexity of any algorithm? Explain with suitable example. [4]

Define Data Structure. Write down the difference between linear and non-linear data structure. [2+3]

Define Data Structure. Write down the difference between linear and non-linear data structure.

Define Omega ana Theta notation with suitable example.

Compare primitive and non-primitive data structures.

Explain about Big-Oh notation with its significance and limitation.

Briefly explain the statement "ADT provides extraction" with a suitable example.

Briefly explain the significance of asymptotic notations with a suitable diagram.

Define data structure with its types.

What do you mean by asymptotic notation? Define Big-O notation with its properties.

Discuss about theta function, Big-Oh function and Omega function.

Define data structure with its importance.

Define big-O notation and big-Ω notation with their respective curves.

Differentiate between primitive and non-primitive data structure.

Define Omega and theta notation with a suitable example.

Define data structure and explain the basic data structure operations.

Define Omega and Theta notation with a suitable example.

Why are data structures needed? Write any data structure as ADT and write applications of stacks..

Can you always insert an item into an empty queue? Explain with possible reasons and examples.

Explain the advantages of dynamic implementation of stack and queue over sequential storage to represent stack and queue.

Define Big 'O' notation and describe the rules to determine the order of common functions.

Compare linear, quadratic, logarithmic, and linear-logarithmic order functions. Compare sequential search and binary search in terms of Big 'O' notation with an example.