PRESENTS:

DATA STRUCTURES

The computer on which this guide is being written can hold up to 122 GB of data. To put things in perspective, roughly 0.052 GB is sufficient to store the collected works of William Shakespeare. Computers hold a lot of data, and ***data structures*** help them make sense of it all. Data structures organize data to make it easier for computers to ***access and manipulate it.*** Let's learn about data structures, and maybe we can free up some storage in our own memories for a few more facts.

By analyzing questions, you can see patterns emerge, patterns that will help you answer questions. Qwiz5 is all about those patterns. In each installment of Qwiz5, we take an answer line and look at its five most common clues. Here we explore five clues that will help you answer a tossup on ***Data Structures.***

Tree

Big (Data) Picture: A tree is a ***nonlinear*** representation of data. This means that the data elements are not arranged in a sequence. Thus, the computer cannot traverse all the elements of a tree in a single run. Instead, if it searches in a linear fashion, it will have to choose a path at the ***nodes*** of the tree to follow on each pass.

Metadata: A ***binary tree*** is a common form of tree with a relatively simple structure. Binary trees are composed of ***nodes***, which in turn are made up of a data element, a right pointer, and a left pointer. Each of these pointers points to a subordinate "subtree." These subtrees are called ***children*** of the ***parent node.*** Although there are other forms of trees, binary search trees are common because they allow computers to perform a rapid ***binary search.***

Running the Numbers: Since trees are a nonlinear data structure, there is more than one way to "visit" (retrieve data from, update, or delete) each node. The process of searching a tree for a given node is known as ***tree traversal*.** This traversal can take the form of a breadth-first or ***level-order*** search (in which all the nodes at a given level are searched before moving to the next level), or a depth-first search (in which you go as far down the tree as possible before searching at the same level).

Data Loss: The amount of time needed to perform an operation on a binary search tree is dependent on the tree's height. If the tree's height is not kept small, operations can take unnecessarily long to perform. Fortunately, binary search trees can be ***self-balancing*** to counter this. Self-balancing binary search trees ***automatically limit a tree's height.***

Buzz On: ***Red-black,*** a special form of self-balancing binary search tree; ***AVL,*** the first self-balancing binary search tree to be invented, named after inventors Adelson-Velsky and Landis.

Arrays

Big (Data) Picture: Arrays, in contrast to trees, are ***linear data structures.*** Arrays store their data contiguously, meaning the data is arranged in sequence, and the position of each datum can be determined by an algorithm.

Metadata: Arrays are the simplest form of data structure, essentially a collection of variables with an associated ***array index.*** The most basic form of array is known as a ***one-dimensional array.*** Two-dimensional arrays are sometimes called ***matrices*** due to their resemblance to the mathematical construct of the same name.

Running the Numbers: Arrays are a flexible form of data storage. The use of an array index can allow quick identification of a desired element from an array. Arrays have another advantage in that their size can be ***dynamically allocated.*** This means that the size of the array can be altered while a search is running based on the amount of data that needs to be stored.

Data Loss: A common problem with arrays stems from a user's failure to perform ***bounds checking.*** Bounds checking determines if a variable "fits" within the range of an array. If a variable that does not fit within the range of acceptable values in an array is placed within that array, serious errors can occur, including ***memory address overwriting.***

Buzz On: ***Stride,*** the distance in memory between variables in an array; ***explode function,*** a function in PHP that breaks a string into an array; ***square brackets,*** in many computer languages arrays are accessed (i.e. information is obtained from them) through the use of square brackets.

Linked List

Big (Data) Picture: Linked Lists are the second form of linear data structure on our guide. Unlike arrays, however, the order of items in a linked list is ***not determined by that item's physical placement.*** Instead, each element ***points*** to the next.

Metadata: Linked Lists consist of ***nodes*** made up of data and a ***pointer.*** The pointer is what directs the sequence to the next element in the data structure.

Running the Numbers: An algorithm's ***time complexity*** is defined as the amount of computer time required to run the algorithm. A Linked List's time complexity takes ***linear time.*** Linear time, usually written ***O(n) (Big O of N),*** means that time complexity increases with the size of the input for a Linked List.

Data Loss: In order to increase the speed of an index, Linked Lists may use ***sentinel nodes***. However, trees may also utilize sentinel nodes as well, so pay attention to the context of the question! Sentinel nodes essentially ***terminate a traversal of the linked list -*** that is, they stop the search.

Buzz On: ***Data structures in Lisp,*** Lisp is a family of programming languages which utilizes the Linked List as their basic data structure.

Stacks

Big (Data) Picture: Stacks are another form of linear data structure. In a stack, data is stored in a sequential order. This order is known as ***LIFO*** (Last in, First Out). LIFO dictates that for any given operation on the stack you can ***only access the element at the top of the stack.***

Metadata: Stacks are ordered lists of similar types of data. The most important things to understand about a stack are its two main operations: ***push*** (inserting a new element on the top of the stack), and ***pop*** (removing an element from the top of the stack).

Running the Numbers: The structure of stacks makes them particularly useful for evaluating ***Reverse Polish Notation.*** Reverse Polish Notation (RPN) is a form of mathematical notation which reverses the typical order of operators and operands. For instance, 5 2 + in RPN is the same as 5 + 2 in standard notation. Using RPN in algorithms saves memory, and stacks, with their *LIFO* structure, are ideally equipped to interpret these algorithms.

Data Loss: Stacks can suffer from a problem known as ***overflow condition.*** When a stack is completely full, according to its parameters, we ***cannot insert any additional values onto it.*** This state, the overflow condition, can only be resolved by removing an item from the stack.

Buzz On: ***Pushdown automaton (PDA),*** a type of automaton (an abstract machine), that employs a stack to make decisions and perform operations; ***Quicker than heap-based allocation,*** Stacks are often contrasted with ***heaps***, another non-linear form of data storage.

Graphs

Big (Data) Picture: A graph represents data in a nonlinear fashion. On the most basic level, a graph is a representation of a set of objects where some of these objects are connected to each other.

Metadata: Graphs are made up of two components: ***vertices*** and ***edges.*** Vertices are the objects and edges are the links that connect them. Vertices that are connected to each other are said to be ***adjacent.*** A ***path*** is the sequence of edges between two vertices.

Running the Numbers: Graphs are ideal data representations for various situations. For instance, let's assume the vertices of a graph represent different rooms at a quiz bowl tournament and edges represent the hallways connecting them. ***Dijkstra's algorithm*** is an extremely useful algorithm for understanding graphs, as it allows a user to ***find the shortest distance between all vertices.*** In the above example, the algorithm could allow us to find the shortest distance between one room (say, the tournament control room) and every other room.

Data Loss: Graphs are quite colorful - literally! Graph coloring refers to the ***assignment of different "colors" to a graph's vertices such that no two vertices have the same color.*** This exercise isn't a trivial one; graph coloring conceptualizes problem solving under constraints and certain restrictions. The ***chromatic number*** of a graph refers to the number of colors needed to successfully perform graph coloring. If a graph is called a ***critical graph*** it can decrease its chromatic number by ***no more than one.*** If more than one vertex or edge is lost and the graph's chromatic number decreases, it can no longer be colored.

Buzz On: ***Floyd-Warshall algorithm,*** an algorithm to find the shortest path between vertices of a weighted graph; ***Seven Bridges of Königsberg problem,*** one of the first mathematical problems utilizing graph theory.

Quizbowl is about learning, not rote memorization, so we encourage you to use this as a springboard for further reading rather than as an endpoint. Here are a few things to check out:

- Want to know a little more about stacks vs. heaps? Try this article that explains the key differences.
- Maybe you've heard of the Seven Bridges of Königsberg problem before, but are a bit confused about the actual concepts. Here's an explanation of the idea behind Euler's solution.
- This article from Medium can help you sort out the Xs and Os of Big O notation.
- What's coming next? Perhaps this article on Quantum Computing and Data Storage might give you some ideas…