

File Version Update:

This version of file contains:

- Content of the Course
- FAQ updated version.(These must be read once because some very basic definition and question are being answered)
- Glossary updated version.(These must be read once because some very basic terms are being explained which you even might not found in the book)
- **Solved Past Assignment Selected for MID Term.** (Coming Soon)
- **Solved Question From Mid-Term Papers** (Coming Soon)
- **MCQs GIGA File (in Compilation)**
(will be shared as Separate File) (Coming Soon)
-

Note: Use Table of Content to view the Topics, In PDF mode , you can check Bookmarks menu. I have tried to provide maximum data on less pages.

----:Table of Content:----

FILE VERSION UPDATE:	1
INTRODUCTION TO DATA STRUCTURE.....	3
COURSE CONTENT:	3
FAQ UPDATED VERSION.(.....	3
QUESTION: DEFINE ABSTRACT BASE CLASS?.....	3
QUESTION: WHAT IS INSTANCE?	3
QUESTION: DEFINE CONCRETE CLASS?	3
QUESTION: NAME THE PROPERTIES OF A BINARY TREE?	3
QUESTION: DEFINE BINARY SEARCH TREES (BST) WITH EXAMPLE?	3
QUESTION: DEFINE HEAP?	4
QUESTION: DEFINE BINARY TREES?	4
QUESTION: HOW CAN WE DIFFERENTIATE AMONG DATABASE, DATA COMMUNICATION AND DATA STRUCTURE?	4
QUESTION: DESCRIBE 32BIT PROGRAMMING AND 16 BIT PROGRAMMING AND DIFFERENCE BETWEEN THEM? IS THERE 64 BIT PROGRAMMING AVAILABLE?	4
QUESTION: WHAT IS DIFFERENCE BETWEEN PSEUDO CODE AND ALGORITHM?	4
QUESTION: WHAT IS EXTERNAL NODE?	4
QUESTION: HOW DATA COMPRESSION WORKS?.....	4
QUESTION: DIFFERENTIATE BETWEEN GREEDY ALGORITHM AND HUFFMAN ALGORITHM?	4
QUESTION: WHAT IS THE FORMULA FOR FINDING THE MINIMUM NUMBER OF NODES IN AVL TREE I.E LOGON FOR	5
QUESTION: WHAT IS DIFFERENCE BETWEEN PAGING AND VIRTUAL MEMORY?	5
QUESTION: WHAT IS TRANSIENT OBJECT?	5
QUESTION: HOW STACK USEFUL IN HARDWARE?	5
QUESTION: WHAT IS GENERIC DATA TYPE?.....	5
QUESTION: DIFFERENCE BETWEEN BINARY, UNARY AND OPERAND?	5
QUESTION: WHY WE CAN'T USE PARENTHESIS AT START IN POSTFIX?	6
QUESTION: POSFIX,INFIX,PREFIX WITH SOME EXAMPLES?	6
QUESTION: DIFFERENCE BETWEEN .H FILE AND .CPP FILE?	6
QUESTION: WHAT IS STANDARD TEMPLATE LIBRARY(STL)?	6
QUESTION: WHAT IS A DEFAULT CONSTRUCTOR?.....	6
QUESTION: WHAT IS DIFFERENCE BETWEEN INT* I AND INT *I?	6
QUESTION: WHAT IS HEAD?	7
QUESTION: WHAT IS ARGV AND ARGV?	7
QUESTION: WHAT IS INLINE FUNCTION IN C++?	7
QUESTION: WHAT IS AN ARRAY?.....	7
QUESTION: WHAT IS AN ARRAY ELEMENT?	7
QUESTION: WHAT IS A POINTER?	7
QUESTION: WHAT IS REAL STORAGE?	7
QUESTION: WHAT IS PRIORITY QUEUE?	7
QUESTION: WHAT IS STACK?.....	7
QUESTION: DIFFERENCE BETWEEN STACK AND QUEUE?	8
QUESTION: WHAT IS METHOD?	8
QUESTION: WHAT IS AN ALGORITHM?.....	8
QUESTION: WHAT IS AN INTERFACE?.....	8
QUESTION: WHAT IS A QUEUE?	8
QUESTION: WHAT IS ABSTRACT DATA TYPE(ADT)?	8
QUESTION: WHAT IS DATA STRUCTURE?	8

QUESTION: WHAT IS MEMBER FUNCTION?..... 8

QUESTION: WHAT IS REFERENCE? 8

GLOSSARY (UPDATED VERSION) 8

SOLVED PAST ASSIGNMENTS (UPDATED VERSION)..... 12

MID-TERM PAST PAPERS (UPDATED VERSION) 12

SOLVED: 12

UNSOLVED: 12

=====

Introduction To Data Structure.

Course Content:

Introduction, the Array data type, the List abstract data type (ADT), Linked lists, Stacks, Queues, Trees, AVL Trees , Huffman encoding , The Heap ADT, Priority queue implementation using the heap ADT, The Disjoint Sets ADT, The Table ADT, implementation using arrays, Skip lists, Table ADT implementation using Hashing, Collision resolution in Hashing, Other uses of Hashing, Sorting, and Selection sort, insertion sort, bubble sort algorithms, Divide and conquer strategy: merge sort, quick sort.

FAQ updated version.(

Question: Define Abstract Base Class?

Answer: An abstract base class is C++ 's way of defining an interface. It is a class which declares that all child classes derived from it must implement a certain set of methods. It is abstract because an instance of the abstract base class may not be created (only concrete implementations of the interface may be instantiated).

Question: What is Instance?

Answer: A class is a definition of a set of data and member functions. When space for the data is actually allocated, we say that a member of the class has been instantiated. The instantiation is called an instance of the class. Each instance has its own set of data (there is also a mechanism in C++ to define data that is only allocated once per class, and shared amongst all instances of the class).

Question: Define Concrete Class?

Answer: A concrete class is a class that is not abstract. An instance of the class may be created, which contains copies of all the data members of the class, and member functions which may be invoked.

Question: Name the Properties of a Binary Tree?

Answer: Since binary tree nodes can only have a maximum of two children, this fact introduces several properties that do not make sense on general trees. There are three important properties that a binary tree can have: fullness, balance, and leftness.

Question: Define Binary Search Trees (BST) with example?

Answer: The most popular variation of the Binary Tree is the Binary Search Tree (BST). BSTs are used to quickly and efficiently search for an item in a collection. Say, for example, that you had a linked list of 1000 items, and you wanted to find if a single item exists within the list. You would be required to linearly look at every node starting from the beginning until you found it. If you're lucky, you might find it at the beginning of the list. However, it might also be at the end of the list, which means that you must search every item before it. This might not seem like such a big problem, especially nowadays with our super fast computers, but imagine if the list was much larger, and the search was repeated many times. Such searches frequently happen on web servers and huge databases, which makes the need for a much faster searching technique much more apparent. Binary Search Trees aim to Divide and Conquer the data, reducing the search time of the collection and making it several times faster than any linear sequential search.

Question: Define Heap?

Answer: 1. A portion of memory reserved for a program to use for the temporary storage of data structures whose existence or size cannot be determined until the program is running. To build and use such elements, programming languages such as C and Pascal include functions and procedures for requesting free memory from the heap, accessing it, and freeing it when it is no longer needed. In contrast to stack memory, heap memory blocks are not freed in reverse of the order in which they were allocated, so free blocks may be interspersed with blocks that are in use. As the program continues running, the blocks may have to be moved around so that small free blocks can be merged together into larger ones to meet the program's needs.

2. A complete binary tree in which the value of any node is not exceeded by the value of either of its children.

Question: Define Binary Trees?

Answer: As the name implies, a binary tree is simply a recursively-defined tree that can have a maximum of two children. The two children are commonly given the names left child and right child, and binary trees have an additional traversal called the in-depth traversal. Binary trees are used for many things, ranging from efficient searching techniques, data compression, and even arithmetic expression parsers (simple compilers). Note that some people refer to binary trees as 'B-Trees', which is incorrect! B-Trees are a special kind of multi-data per node tree used for advanced balancing algorithms, and Apple Macintoshes even use B-trees for their file systems. Be careful to not call a 'Binary Tree' a 'B-Tree'!

Question: How can we differentiate among database, data communication and data structure?

Answer: Database: A collection of data arranged for ease and speed of search and retrieval.
Data Communication: Communications involves data transfer from one computer to another through a communications medium, such as a telephone, microwave relay, satellite link, or physical cable. Two primary methods of computer communications exist: temporary connection of two computers through a switched network, such as the public telephone system, and permanent or semipermanent linking of multiple workstations or computers in a network. The line between the two is indistinct, however, because microcomputers equipped with modems are often used to access both privately-owned and public-access network computers.

Data Structure: An organizational scheme, such as a record or array, that can be applied to data to facilitate interpreting the data or performing operations on it.

Question: Describe 32bit programming and 16 bit programming and difference between them? Is there 64 bit programming available?

Answer: 16 bits programming was done using 16-bits registers which is now obsolete. Now 32-bit programming is done using 32-bits registers which is now a days used for constructing windows application. These are the cases of IBM compatible PCs which consists of 32 bits register until now. There are other architecture which are consists of 64 bits registers e.g. Sun Sparc Machines are 64 bits Machines. So it depends upon the architecture of the Machine and softwares are written accordingly.

Question: What is difference between Pseudo code and Algorithm?

Answer: Pseudo code: Any informal, transparent notation in which a program or algorithm description is written. Many programmers write their programs first in a pseudocode that looks much like a mixture of English and their favorite programming language, such as C or Pascal, and then translate it line by line into the actual language being used.

Algorithm: A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.

Question: What is external node?

Answer: A terminal or "bottom" item of a tree, i.e. an item with no child known as external node.

Question: How Data Compression Works?

Answer: The goal of data compression is to represent an information source (e.g. a data file, a speech signal, an image, or a video signal) as accurately as possible using the fewest number of bits.

Question: Differentiate between Greedy Algorithm and Huffman Algorithm?

Answer: Greedy Algorithm: An algorithm that always takes the best immediate, or local, solution while finding an answer. Greedy algorithms find the overall, or globally, optimal solution for some optimization problems, but may find less-than-optimal solutions for some instances of other problems.

Huffman Algorithm: A method of compressing a given set of data, based on the relative frequency of the individual elements. The more often a given element, such as a letter, occurs, the shorter, in bits, is its corresponding code. It was one of the earliest data compression codes and, with modifications, remains one of the most widely used codes for a large variety of message types.

Question: What is the formula for finding the minimum number of nodes in AVL tree i.e log₂ n for BST?

Answer:

AVL Trees

similar to binary search trees

difference: for every node left and right subtrees can have height difference of at most 1

height of an AVL tree = at most $1.44 \log n$ (approx.)

$N(h)$: # of nodes in minimum size AVL tree of height h

$N(h) = N(h-1) + N(h-2) + 1$

$N(0) = 1, N(1) = 2$

all operations, except insertion, performed in $O(\log N)$ time.

Question: What is difference between Paging and Virtual Memory?

Answer: Paging: The transfer of pages of data between a computer's main memory and an auxiliary memory.

Virtual Memory: Computer memory, separate from the main memory of a specific machine that can be used as an extension of the machine's main memory

Question: What is transient object?

Answer: A transient object is an instance of an object type. Its lifetime cannot exceed that of the application. The application can also delete a transient object at any time.

Question: How Stack useful in hardware?

Answer: Stack data structure has a large number of uses:

- 1) Activation records for function calls are stored on a stack.
- 2) Eliminating recursion to make an algorithm faster can be achieved by using a stack.
- 3) Stacks can be used in expression evaluation
- 4) Stacks are used by compilers during parsing.
- 5) Search algorithms often exploit stack structures.
- 6) It helps you in memory management.

Question: What is generic data type?

Answer: A generic data type is a type whose complete specification is deferred until it is actually used in one or the other way. The missing pieces must be specified when used. This is what C++ knows as "templates".

Question: Difference between binary, unary and operand?

Answer:

Binary Operator: Any Mathematical operation which involves two operands for one operator e.g. $A+B$

Where A and B are operands and $+$ is a binary operator.

Unary Operator: Characteristic of a mathematical operation with a single operand (object)

Operand: A quantity on which an operation is performed.

Question: Why we can't use parenthesis at start in postfix?

Answer: Parenthesis are needed in infix expression only because that is the only way to override the default precedence of operators. In C++, if you write

```
x = 4 + 3 * 2;
```

x will get 10. If you really meant to say add 3 to 4 and then multiply by 2, you will need to write

```
x = (4+3) * 2;
```

The parenthesis are not necessary in postfix form because a binary operator appears after its two operands

Question: Postfix, Infix, Prefix with some Examples?

Answer: Postfix: In postfix notation, the operator is written after the operand/s it operates on.

Infix: In infix notation, the operator is written in between the operands it operates on as in the case of binary operators.

Prefix: In prefix notation, the operator is written before the operand/s it operates on.

Exmples of Infix to Postfix:

Infix	Postfix
A + B	AB +
12 + 60 - 23	12 60 + 23 -
(A + B) * (C - D)	AB + C D - *
A * B * C - D + E / F	AB * C * D - E F / +

Question: Difference between .h file and .cpp file?

Answer: Header File: The subdirectory called INCULDE contains header files. These files (also called "include" files) are text files, like the ones you generate with a word processor or the Dev C++ Editor. Header files can be combined with your program before it is compiled, in the same way that a typist can insert a standard heading in a business letter. Each header file has .h file extension. Header files serve several purposes. You can place statements in your program listing that are not program code but are instead messages to the compiler. These messages, called compiler directives, can tell the compiler such things as the definitions of words or phrases used in your program. Some useful compiler directives have been grouped together in header files, which can be included in the source code of your program before it goes to the compiler.

CPP File: Class methods are defined (implement) in the CPP files. Also main function is defined in the CPP file.

Question: What is Standard Template Library(STL)?

Answer: At its July 1994 meeting, the ANSI/ISO C++ Standards Committee voted to adopt STL as part of the standard C++ library. The STL proposal to the committee by Alex Stepanov and Meng Lee of Hewlett-Packard Labs was based on research on generic programming and generic software libraries that Stepanov, Lee, and David Musser have been working on for several years, in Scheme, Ada, and C++. The Standard Template Library a library of reusable containers and is now part of the C++ Standard Library. The STL provides C++ programmers with a library of common data structures --linked lists, vectors, deques, sets, and maps -- and a set of fundamental algorithms that operate on them.

Question: What is a default constructor?

Answer: A constructor that doesn't take any parameters. The following class only has one constructor, it doesn't take parameters, and therefore is a default constructor. `class date { public: date() { cout << "This is a DEFAULT constructor" << endl; } };` The term default is used because when an object is created without passing any parameters to the constructor, the default is used.

Question: What is difference between int* i and int *i?

Answer: There is no difference between int* i or int *i as far as the compiler is concerned. Both declare 'i' to a pointer to an integer. The actual difference is how this declaration is perceived. The declaration syntax in general is "type variable", e.g., "int x", "double z", "char c" etc. If we follow this pattern, suppose we want the type of a variable "time" to be "pointer to int". To do so, we would write "int* time"; "int*" is the type and "time" is variable. However, the notation "*time" can be thought of as a variable name with the "*" included. [Variables names cannot being with a "*" normally]. To get to the actual integer

stored in memory, the syntax is `"*time"`, e.g., `"*time = 10"`, `"y = *time"`. In this approach, the type is `"int"` and the variable name is `"*time"` and the declaration thus `"int *time"`

Question: What is head?

Answer: Head: The first item of a list is called head.

Question: What is argc and argv?

Answer: C and C++ have a special argument list for `main()`, which looks like this:

```
int main(int argc, char* argv[]) { // ...
```

The first argument is the number of elements in the array, which is the second argument. The second argument is always an array of `char*`, because the arguments are passed from the command line as character arrays (and remember, an array can be passed only as a pointer). Each whitespace-delimited cluster of characters on the command line is turned into a separate array argument. The following program prints out all its command-line arguments by stepping through the array:

Example:

```
#include
int main(int argc, char* argv[]) {
cout << "argc = " << argc << endl;
for(int i = 0; i < argc; i++)
cout << "argv[" << i << "] = "
<< argv[i] << endl;
}
```

You'll notice that `argv[0]` is the path and name of the program itself. This allows the program to discover information about itself. It also adds one more to the array of program arguments, so a common error when fetching command-line arguments is to grab `argv[0]` when you want `argv[1]`.

You are not forced to use `argc` and `argv` as identifiers in `main()`; those identifiers are only conventions (but it will confuse people if you don't use them). Also, there is an alternate way to declare `argv`:

```
int main(int argc, char** argv) { // ...
```

Both forms are equivalent

Question: What is inline function in C++?

Answer: In programming, referring to a function call replaced with an instance of the function's body. Actual arguments are substituted for formal parameters. An inline function is usually done as a compile-time transformation to increase the efficiency of the program. Using inline functions can reduce execution time but increase program size.

Question: What is an Array?

Answer: An array is a data structure that allows storage of a sequence of values. The values are stored in a contiguous block of memory. Arrays allow fast random access to particular elements. If the number of elements is indefinite or if insertions are required then we can't use an array. Example:

```
int idnumbers[100];
```

This declares an array of 100 integers named `idnumbers`.

Question: What is an Array Element?

Answer: A data value in an array

Question: What is a Pointer?

Answer: In programming and information processing, a variable that contains the memory location (address) of some data rather than the data itself.

Question: What is Real Storage?

Answer: The amount of RAM memory in a system, as distinguished from virtual memory. Also called physical memory, physical storage.

Question: What is Priority Queue?

Answer: A priority queue is a specialized queue in which the items are stored in order. A priority queue allows access to the smallest (or sometimes the largest) item.

Question: What is Stack?

Answer: A collection of items in which only the most recently added item may be removed. The latest added item is at the top. Basic operations are push and pop. Also known as "last-in, first-out" or LIFO.

Question: Difference between Stack and Queue?

Answer: Stack is "Last in first out" LIFO. Queue is "First in first out" FIFO

Question: What is Method?

Answer: A Method, or Member Function is a routine which is associated with a data structure to make a class. A Method can be invoked, and when it executes it has access to the data in the class, as well as data passed by way of arguments.

Question: What is an Algorithm?

Answer: An Algorithm is a generic term for any procedure. We usually mean a procedure that can be implemented as a routine, and which performs some well defined task. In general there are several possible Algorithms to perform the same task. NAO's Algorithm class is an organizational class that does not define any interface, but merely groups classes. Algorithms that perform the same task (or type of task) will share an interface defined in an abstract base class derived from the Algorithm class.

Question: What is an Interface?

Answer: The interface of a class is the set of routines which are provided to manipulate instances of the class.

Question: What is a Queue?

Answer: A multielement data structure from which (by strict definition) elements can be removed only in the same order in which they were inserted; that is, it follows a first-in-first-out (FIFO) constraint. The important queue operation are inserting an item at the rear of the queue and removing the item from the front of the queue.

Question: What is Abstract Data Type(ADT)?

Answer: In programming, a data set defined by the programmer in terms of the information it can contain and the operations that can be performed with it. An abstract data type is more generalized than a data type constrained by the properties of the objects it contains—for example, the data type "pet" is more generalized than the data types "pet dog," "pet bird," and "pet fish." The standard example used in illustrating an abstract data type is the stack, a small portion of memory used to store information, generally on a temporary basis. As an abstract data type, the stack is simply a structure onto which values can be pushed (added) and from which they can be popped (removed). The type of value, such as integer, is irrelevant to the definition. The way in which the program performs operations on abstract data types is encapsulated, or hidden, from the rest of the program. Encapsulation enables the programmer to change the definition of the data type or its operations without introducing errors to the existing code that uses the abstract data type. Abstract data types represent an intermediate step between traditional programming and object-oriented programming.

Question: What is Data Structure?

Answer: A data structure is a way of grouping fundamental types (like integers, floating point numbers, and arrays) into a bundle that represents some identifiable thing. For example, a matrix may be thought of as the bundle of the number of rows and columns, and the array of values of the elements of the matrix. This information must be known in order to manipulate the matrix. C introduced the struct for declaring and manipulating data structures. C++ extended the struct to a class.

Question: What is Member Function?

Answer: A Member Function, or Method is a routine which is associated with a data structure to make a class. A Member Function can be invoked, and when it executes it has access to the data in the class, as well as data passed by way of arguments.

Question: What is Reference?

Answer: To access a variable, such as an element in an array or a field in a record.

Glossary (Updated Version)

Abstract Data Type : A set of data values and associated operations that are precisely specified independent of any particular implementation. Also known as ADT

Algorithm : A computable set of steps to achieve a desired result.

Alias : An alternative name for the same object. A "nickname".

Ancestor : A parent of a node in a tree, the parent of the parent, etc.

Argument : A value passed to a called function by the calling function.

Array : In programming, a list of data values, all of the same type, any element of which can be referenced by an expression consisting of the array name followed by an indexing expression. Arrays are part of the fundamentals of data structures, which, in turn, are a major fundamental of computer programming

AVL tree : A balanced binary search tree where the height of the two subtrees (children) of a node differs by at most one.

Balanced Binary Tree : A binary tree where no leaf is more than a certain amount farther from the root than any other. After inserting or deleting a node, the tree may be rebalanced with "rotations."

big-O notation : A theoretical measure of the execution of an algorithm, usually the time or memory needed, given the problem size n , which is usually the number of items. Informally, saying some equation $f(n) = O(g(n))$ means it is less than some constant multiple of $g(n)$. The notation is read, "f of n is big oh of g of n".

Binary Heap : A complete binary tree where every node has a key more extreme (greater or less) than or equal to the key of its parent.

Binary Search : A type of search algorithm that seeks an item, with a known name, in an ordered list by first comparing the sought item to the item at the middle of the list's order. The search then divides the list in two, determines in which half of the order the item should be, and repeats this process, until the sought item is found.

Binary Search Tree : A data structure with in which every node refers to a left subtree and a right subtree such that all values in the left subtree are smaller than the value in the node and all elements in the right subtree are greater than (or equal to) the value in the node. The top node is called the root. The nodes with no children (left and right subtrees empty) are called leaves.

Binary Tree : A specific type of tree data structure in which each node has at most two subtrees, one left and one right. Binary trees are often used for sorting information; each node of the binary search tree contains a key, with values less than that key added to left subtree and values greater than that key added to the right subtree.

Bounded Queue : A queue limited to a fixed number of items.

Bubble Sort : Sort by comparing each adjacent pair of items in a list in turn, swapping the items if necessary, and repeating the pass through the list until no swaps are done

Build-Heap : Convert an array into a heap by executing heapify progressively closer to the root. For an array of n nodes, this takes $O(n)$ time under the comparison model.

Child : An item of a tree referred to by a parent item. Every item, except the root, is the child of some parent.

Circular List : A linked list in which the rear item refers back to the head item

Circular Queue : An implementation of a bounded queue using an array.

Collision : When two or more items should be kept in the same location, especially in hash tables, that is, when two or more different keys hash to the same value.

Collision Resolution Scheme : A way of handling collisions, that is, when two or more items should be kept in the same location, especially in a hash table. The general ways are keeping subsequent items within the table (open addressing), keeping a list for items which collide (direct chaining hashing or separate chaining hashing), keeping a special overflow area, etc. Perfect hashes avoid collisions, but may be time-consuming to create.

Complete Binary Tree : A complete binary tree of depth d is the strictly binary tree all of whose leaves are at level d

Data Object : An object capable of storing data. A variable or a constant. (A function is allocated memory within the computer and is therefore an object; but it is not a data object because it cannot store data).

Data Structure : The term data structure refers to the way data is organized for use within a program. Correct organization of data can lead to simpler and more efficient algorithms. Common data structures are linked-lists, stacks, queues and trees.

Descendant : A child of a node in a tree, any of the children of the children, etc.

disjoint Set : A set whose members do not overlap, are not duplicated, etc.

Doubly Linked List : A data structure in which each element contains pointers to the next and previous elements in the list, thus forming a bidirectional linear list.

Dynamic Array : An array whose size may change over time. Items are not only added or removed, but memory used changes, too.

FIFO : First in first out is a policy that items are processed in order of arrival. A queue implements this.

full binary tree : A binary tree in which each node has exactly zero or two children.

Hash Function : A function that maps keys to integers, usually to get an even distribution on a smaller set of values.

Hash Table : A dictionary in which keys are mapped to array positions by a hash function. Having more than one key map to the same position is called a collision. There are many ways to resolve collisions, but they may be divided into open addressing, in which all elements are kept within the table, and chaining, in which additional data structures are used.

head : The first item of a list.

Heap : An area of memory from which space for dynamic structures are allocated

heap property : Each node in a tree has a key which is more extreme (greater or less) than or equal to the key of its parent.

heap. : A complete tree where every node has a key more extreme (greater or less) than or equal to the key of its parent. Usually understood to be a binary heap.

Heapify : Rearrange a heap to maintain the heap property, that is, the key of the root node is more extreme (greater or less) than or equal to the keys of its children. If the root node's key is not more extreme, swap it with the most extreme child key, then recursively heapify that child's subtree. The child subtrees must be heaps to start.

Height : The maximum distance of any leaf from the root of a tree. If a tree has only one node (the root), the height is zero.

Huffman encoding : A minimal variable-length character encoding based on the frequency of each character. First, each character becomes a trivial tree, with the character as the only node. The character's frequency is the tree's frequency. The two trees with the least frequencies are joined with a new root which is assigned the sum of their frequencies. This is repeated until all characters are in one tree. One code bit represents each level. Thus more frequent characters are near the root and are encoded with few bits, and rare characters are far from the root and are encoded with many bits.

In-order Traversal : Process all nodes of a tree by recursively processing the left subtree, then processing the root, and finally the right subtree.

Infix Notation : A notation in which operators appear between the operands, as in $3 + 5$

Insertion Sort : Sort by repeatedly taking the next item and inserting it into the final data structure in its proper order with respect to items already inserted.

Instance : A class is a definition of a set of data and member functions. When space for the data is actually allocated, we say that a member of the class has been instantiated. The instantiation is called an instance of the class. Each instance has its own set of data (there is also a mechanism in C++ to define data that is only allocated once per class, and shared amongst all instances of the class).

Internal Node : A node of a tree that has one or more child nodes, equivalently, one that is not a leaf

Key : The part of a group of data by which it is sorted, indexed, cross referenced, etc.

Leaf : Any node (location) in a tree structure that is at the farthest distance from the root (primary node), no matter which path is followed. Thus, in any tree, a leaf is a node at the end of a branch—one that has no descendants.

left rotation : In a binary search tree, pushing a node N down and to the left to balance the tree. N's right child replaces N, and the right child's left child becomes N's right child.

Level-order Traversal : Process all nodes of a tree by depth: first the root, then the children of the root, etc.

LIFO : Last in first out is a policy that the most recently arrived item is processed first. A stack implements this.

Linear Search : A simple, though inefficient, search algorithm that operates by sequentially examining each element in a list until the target element is found or the last has been completely processed. Linear searches are primarily used only with very short lists. Also called sequential search.

Linked List : A data structure in which a list of nodes or elements of a data structure connected by pointers. A singly linked list has one pointer in each node pointing to the next node in the list; a doubly linked list has two pointers in each node pointing to the next and previous nodes. In a circular list, the first and last nodes of the list are linked together.

max-heap property : Each node in a tree has a key which is less than or equal to the key of its parent.

Merge : Combine two or more sorted sequences of data into a single sorted sequence.

merge sort : A sort algorithm that splits the items to be sorted into two groups, recursively sorts each group, and merges them into a final, sorted sequence.

min-heap property : Each node in a tree has a key which is greater than or equal to the key of its parent.

Name Tag : A name tag in C++ is a set of text characters formed into a symbolic word used to refer to an object. Name tags must start with an alpha character or an underscore. The second or subsequent characters can be alpha or numeric characters or the underscore character. All other characters are not

allowed. Capital alpha characters can be used and are interpreted by C++ as different to their lower case equivalents.

Node : (1) A unit of reference in a data structure. Also called a vertex in graphs and trees. (2) A collection of information which must be kept at a single memory location.

null tree : A tree which is empty.

Object : Any program entity which uses physical memory in the computer

Object Oriented Programming : A concept of programming in which elements of the program are coded as stand alone objects. Each object is completely self contained in that it incorporates methods whereby the object can manipulate its own characteristics. A "Door" object, for instance would know how to open and close itself. It would also be able to respond to interrogation and advise the enquirer whether it is currently open or closed.

Open Addressing : A general collision resolution scheme for a hash table in which all items are stored within the table. In case of collision, other positions are computed and checked (a probe sequence) until an empty position is found. Some ways of computing possible new positions are less efficient because of clustering.

Overload : A term used to refer to the use of one symbol for more than one purpose. For instance, in mathematics the "-" symbol is used both as a negation symbol and as a subtraction symbol. In C++ the "<<" symbol is used as an output operator and as a shift left operator. Functions which implement a new operation for a previously used operator are called operator overload functions. Different functions which have the same function name are called overloaded functions.

Parameter : A value received by a called function from a calling function.

perfect binary tree : A binary tree with all leaf nodes at the same depth. All internal nodes have degree 2.

Postorder Traversal : Process all nodes of a tree by recursively processing the left subtree, then processing the right subtree, and finally the root.

Prefix Notation : A notation in which operators appear before the operands, as in +A B

Preorder Traversal : Process all nodes of a tree by recursively processing the root, then processing the left subtree, and finally the right subtree.

Queue : A data structure with first-in first-out behavior, supporting the operations enqueue (to insert) and dequeue (to remove)

Quicksort : An in-place sort algorithm that uses the divide and conquer paradigm. It picks an element from the array (the pivot), partitions the remaining elements into those greater than and less than this pivot, and recursively sorts the partitions

recursion : An algorithmic technique where a function, in order to accomplish a task, calls itself with some part of the task.

right rotation : In a binary search tree, pushing a node N down and to the right to balance the tree. N's left child replaces N, and the left child's right child becomes N's left child.

Root : The distinguished initial or fundamental item of a tree. The only item which has no parent

Rotation : To switch children and parents among two or three adjacent nodes to restore balance to a tree.

run time : The amount of time needed to execute an algorithm.

Selection Sort : A sort algorithm that repeatedly looks through remaining items to find the least one and moving it to its final location. The run time is (n^2) , where n is the number of comparisons. The number of swaps is $O(n)$.

sibling : A node in a tree that has the same parent as another node is its sibling.

Singly Linked List : A data structure in which a list of nodes or elements of a data structure connected by pointers. A singly linked list has one pointer in each node pointing to the next node in the list

Sort : Arrange items in a predetermined order. There are dozens of algorithms, the choice of which depends on factors such as the number of items relative to working memory, knowledge of the orderliness of the items or the range of the keys, the cost of comparing keys vs. the cost of moving items, etc.

Stack : A data structure with last-in first-out behavior, supporting the operations push (to insert) and pop (to remove)

Strictly Binary Tree : A binary tree is said to be a strictly binary tree if every non-leaf node in a binary tree has non-empty left and right subtrees.

String : A list of characters, usually implemented as an array. Informally a word, phrase, sentence, etc. Since text processing is so common, a special type with substring operations is often available.

Structure : A mechanism which allows objects of different types to be grouped together as a single compound type.

tail : The last item of a list.

threaded tree : A binary search tree in which each node uses an otherwise-empty left child link to refer to the node's in-order predecessor and an empty right child link to refer to its in-order successor.

Tree : A data structure containing zero or more nodes that are linked together in a hierarchical fashion

Tree Traversal : A technique for processing the nodes of a tree in some order.

uniform hashing : A conceptual method of open addressing for a hash table. A collision is resolved by putting the item in the next empty place given by a probe sequence which is independent of sequences for all other key.

Union : The union of two sets is a set having all members in either set.

vertex : An item in a graph. Sometimes referred to as a node.

worst case : The situation or input that forces an algorithm or data structure to take the most time or resources.

Solved Past Assignments (Updated Version)

Mid-Term Past Papers (Updated Version)

Solved:

Unsolved: