

Review for Final Exam

This review covers the second half of the semester after the midterm exam. Together with the midterm review it provides a comprehensive outline of the topics covered in the semester.

Chapter 7 Trees

- Tree: non-linear container class for storing linear data or hierarchical data.
- Full tree, complete tree
- Trees can be stored using a linked structure, can also be stored in an array by index.
- Binary tree: rooted tree in which each node has at most two children. If a binary tree is stored in an array with root at index 1, then for every node with index k , the left child is stored at index $2k$ and the right child is stored at index $2k + 1$. The parent of the node stored at index k is stored at index $k//2$.
- binary search tree: for every node, the value of the data at each node in the left subtree is less than or equal to the value of the data at the node, and the value of the data at each node in the right subtree is greater than or equal to the value of the data at the node. A binary search tree can be implemented to support $\Theta(\log n)$ search, insertion, and deletion operations.
- Tree algorithm are of written recursively since the tree has a recursive structure.
- The three common binary tree traversals are preorder, in-order, and postorder. An in-order traversal of a binary search tree produces the items sorted into increasing order.

Chapter 8 Intro to C++

- Compiling and linking: C++ source code is compiled to machine instructions. Then already existing machine code from libraries is linked with the compiled version of the source code to create executable machine code. There is a preprocessor step before the compile step in which the content of other files is inserted into the C++ source code.
- Syntax items: comments, Blocks { }, semicolon ; for end of statement, string enclosed in double quotes, single character enclosed in single quotes, newline character '\n', tab character '\t'
- Console input stream cin, console output stream cout
- Include statement, namespaces

- Lots of keywords with special meanings: and, bool, namespace, struct, using, break, const, double, if new, private, short, typedef, false, inline, protected, template, void, else, float, int, public, this, char, for, operator, static, class, delete, friend, or, true, while – and more
- C++ variable: name for a memory location that stores a value. C++ variables must be declared with datatype before being used. Basic datatypes are int, char, float, double, bool. Legal names or identifiers must start with letter or underscore and can contain letters, digits, and underscores.
- C++ operators and operator precedence
- Control structures: if, if/else, for, while, do while
- C++ arrays store groups of data of the same type (homogeneous) in contiguous memory accessed by indexing with [].
- Parameters to C++ functions can be passed by value or passed by reference(&). With pass-by-value, the function is given a copy of the parameter's value. With pass-by-reference, the formal parameter in the function refer to the same memory location as the actual parameter. With pass-by-reference, a function can change the value of an actual parameter.
- Header files are used to declare functions and global variables. Short function definitions can be placed in the header file, with definition preceded by keyword inline.
- The scope of a variable is the section of code in which the variable is accessible. The lifetime of a variable is the period of execution of the program in which a memory location is bound to the variable name. A variable can be declared static; its lifetime will the entire period of execution of the program.

Chapter 9 C++ Classes

- C++ has classes for strings in <string>: classname string, methods length, empty, append, insert, and more
- C++ has classes for file input and output in <fstream>. ofstream is output file stream. Declare the object, use open method to attach to a file to write to, write to object using << operator, close. ifstream is input file stream. Declare the object, use open method to attach to a file to read from, read from object using >> operator, close.
- C++ classes are written in two files usually: class definition in header file, implementation of methods in implementation file.
- C++ constructors have the same name as the class and have no return type. They are called automatically when a variable is defined.
- Operators can be overloaded to work with objects in a class. Some must be overloaded as non-members since the calling object is the object to the left of the operator.

Chapter 10 Dynamic memory

- C++ pointers can be used to allocate memory for an array at run time using the new statement.
- Memory allocated using new must be deallocated when no longer needed using delete.
- Memory leak occurs when you allocate memory with new operator and never deallocate it. If your program does this, it will occupy more memory than the program needs, and cause the computer to slow down.
- C++ pointers are used to allocate memory for an array at run-time.
- C++ class for dynamic array allows for the automatic resizing of an array when more memory is needed. The class needs a resize method.
- Classes that use dynamic memory allocation must implement a destructor to deallocate any memory used by a class instance when the instance goes out of scope. The destructor will be called automatically. Such classes must also either write a copy constructor and overload assignment = to make a deep copy or declare these methods private.
- When using pointers, it is possible for a program to access memory that is not allocated for that program and cause the program to crash with a segmentation fault.

Chapter 11 C++ Linked Structures

- C++ class for linked list uses a pointer to link one node in the list to the next node.
- C++ linked list is implemented in a similar fashion to the Python class for linked list.
- typedef to create an alias for a typename. Then code can be made to work for different types by changing the typedef statement.

Chapter 12 C++ Templates

- Template functions can be defined to write functions that work for any type, if the operations used in the function definition are defined for the type. Precede the function prototype by: template <typename Item> and use Item for the typename.
- The compiler generates machine instructions for a template function when the function is called with a specific data type. The machine instructions depend on the data type.
- Template classes work in the same fashion. Use: template <class T> before the class definition. T stands in for a typename.
- C++ has a library – Standard Template Library or STL – with template versions of vector (dynamic array), list (doubly linked list), queue (FIFO queue), priority queue, stack (LIFO stack), set.

Chapter 13 Heaps and Hash Tables

- Binary heap: complete binary tree with property that for each node, data at that node is greater than or equal to data at each of its children.
- Binary heap can be used to implement a priority queue, and to sort items in increasing order in place
- Binary heap can be stored in an array.
- Basic operators for binary heap class are insert: insert a new method at the leftmost available position, and move it up the tree until the heap property is satisfied, and remove_max: remove the largest item from the root, move the last item to the root, and then push that item down the tree until the heap property is satisfied.
- Binary heap operation to move an item down the tree until the heap property is satisfied is called heapify.
- Operations insert and remove max are time $\Theta(\log n)$.
- By using remove_max repeatedly, the data of the heap can be sorted into increasing order in place in the same array in time $\Theta(n \log n)$.
- Hash table: data structure that stores (key, value) pairs. The pairs are stored in an array at an index that is computed by applying a function h hash function – to the key to create an index. They provide $\Theta(1)$ time for insert, lookup, and delete operations typically, but the worst case can be $\Theta(n)$.
- A collision happens when two different keys map to the same index by the hash function. Solutions to collisions are chaining: store multiple (key, value) pairs in a list, and open addressing: use another function to find another index to store the (key, value) pair.
- Hash functions should minimize collisions as much as possible.

Exam structure:

Part I: paper and pencil, closed notes and book, no use of computer, multiple choice, true-false, short answer
Covering definitions and concepts, programming techniques
Part II: open book, open notes, using computer, programming exercises