Review test 2 - roper 235

What order is used in the C++ tool chain?

* Preprocessor, compiler, assembler, linker

C++ source files are nnnnn into one text file by :

* merged, Linker( handles #include\_\_\_\_ and .h files)

What is a template class?

* you can use it for diff object types over and over  
  contained entirely in the header
* facilitates the same set of operations on a variety of data types.
* C++ uses a class template as a pattern to create a new class of a specified type.

What is the proper relationship between class List and class Ordered\_List?

* class Ordered\_List has-a List
* list is a linked list  
  ordered list is a linked list based on comparison to each other (will only have insert)

The nnnnn nnnnn compilation directives \_\_\_:

* pre-processor conditional, are used to eliminate lines at compilation time

Chapter 5 – Stacks

Which of the following accurately describes the behavior of a stack?

* Last in first out

Suppose you want to access the nnnnn element pushed on a stack. Which function would you most likely use?

* Push(), pop(),
* NEXT element pushed on a stack, top()

Of the following applications, which is best suited for a stack?

* Conversions of post fix to infix, rail road

Basic stack operations include

* Push(), pop(), top()

Why doesn't it make sense to have nnnnn for stack?

* The order it comes on vs the order it comes off will be different

What is the resulting output of the program below?

To check for nnnnn nnnnn, use a stack of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

* Balanced parentheses, use open parentheses

The two general methods used to implement a Stack ADT are

* Stack itself, or a wrapper stack
* Use contiguous structure or linked structure

Converting INFIX to POSTFIX requires a stack of OPERANDS. INFIX-TO-POSTIFIX uses stak of OPERATORS

POSTFIX evaluation a stack of OPERANDS

POSTFIX-TO-INFIX a stack of OPERATORS

What is the result of the postfix expression:

A + B \* C + D - ++ A \* BCD - ABC\*+D+

(A+B)\*(C+D). - \*+AB+CD. - AB+CD+\*

A\*B+C\*D. - +\*AB\*CD. - AB\*CD\*+

A+B+C+D. - +++ABCD. - AB+C+D+

Chapter 6 – Queues and Deques

What is the resulting output of the program below

· A Queue ADT can be implemented with a

* Calls for a circular array

· Which of the following most accurately describes the behavior of a queue?

* First in first out

· Which of following the most accurately compares a stack and a queue?

* A queue differs from a stack in only one operation

· The contiguous implementation of a Queue ADT

* Calls for a circular array
* Is more like the contiguous impelentation of ADT Deque

· The reallocate function for class Queue

* Does not copy elements to identical index positions
* Reallocates more memory

· Which representation of a queue has the most favorable Big-O’s for the operations (as discussed in class)?

* A circle (just change index instead of making new node

· If a Queue ADT implemented with an nnnnn becomes full, the queue should

* Array representation, call a reallocate function specialized for the array-based queue

· The contiguous implementation of a Deque ADT is

* More like the ADT QUEUE

· If you need to provide a stack, a queue, an input-restricted deque, and/or an output restricted deque,

* Each of the others should “have” a deque as a data member

· For the contiguous representation of a Deque ADT, when an nnnnn of the deque is reached, WRAP AROUND THE OTHER END.

Chapter 7 – Recursion

· What is recursion?

A problem solving approach that causes functions to call themselves

· What is the resulting output of the program below?

· What nnnnn are required for a function to be recursive?

A base case and a recursive call

· What happens when a recursive function calls itself?

A new instance of the function is called

· When a recursive function returns after X ITERATIONS iterations because it encounters a base case, where does it return to? THE PREVIOUS ITERATION.

· Recursion typically is justified when it reduces DESIGN AND CODING TIME

· What is the downside of recursion?

MAY RESULT IN RUN TIME ERROR, WHEN NO MORE MEMORY AVAILABL.

· Which of the following functions is the nnnnn efficient when implemented recursively? LEAST - CALC THE NTH FIBONACCI ERROR

· In tail recursion, THE RECUSRIVE CALL IS THE LAST EXECUTABLE STATEMENT IN THE FUNC

· With the recursive linear search algorithm given in the book, how many items in the list need to be checked if the item is not in the list? EVERY ITEM IN THE LIST

· What makes backtracking different from random trial and error?

IT IS A SYSTEMATIC APPROACH THAT WILL NOT TRY THE SAME PATH MORE THAN ONCE

· What are the maze backtracking algorithm base cases?

THERE ARE THREE: COORDINATE IN THE MAZE, AM I ON THE WALL, AM I AT THE END

· If there is more than nnnnn possible path through a maze, the maze backtracking algorithm WILL FUNCTION PROPPERLY

Chapter 8 – Trees

· Pick the most correct binary tree assertion.

ALL NODES HAVE 2 OR 0 CHILDREN

· Compared with a singly linked list, a binary tree:

HAS NODES WITH TWO CHILD NODES INSTEAD OF ONE

· Searching for an item in a linked list is O(\_\_\_). Searching in a binary search tree is O(\_\_\_). N, LOG(N)

· How are trees and binary trees related?

BINARY TREE IS A TREE

· Of the following applications, which would be best implemented with a nnnnn tree?

NOT A FAMILY TREE

· Traversal algorithms are relatively simple. What nnnnn of trees makes that so?

INORDER (LEFT, ROOT, RIGHT)

PREORDER (ROOT, LEFT, RIGHT)

POSTORDER (LEFT, RIGHT, ROOT)

SEE ADDITIONAL NOTES FOR DETAILS

· The in-order traversal sequence of a binary tree is nnnnn. The root is

· If a particular binary tree is full and has nnnnn leaf nodes, how many internal nodes does it have?

INTERNAL NODES = LEAF NODES - 1

· If the number nnnnn is inserted into the following binary tree, what would be the nnnnn traversal order?

SEE ADDITIONAL NOTES FOR DETAILS

· If the number nnnnn is deleted from the above tree, what would be the level-order traversal order?

LEVEL ORDER(ROOT, CHILDREN) OUT PUT BY LEVEL LEFT TO RIGHT

· Suppose you have a binary search tree. Which kind of traversal would print the contents in nnnnn order?

INCREASING? IN ORDER

· A heap REQUIRES EVERY SUBTREE TO BE A HEAP

IS A TREE

\*If the size of a heap is nnnnn, the performance

BEST: HEAP IS *N,* O(N^2)

SECOND BEST: 1N, LOG(N) FOR INSERTION

THIRD BEST: N, O(LOGN) FOR INSERTION

OTHERS:

N, O(N^2 FOR RESTRUCTURING

INSERT: O(1), SEARCH: O(N), DELETE: O(LOG(N)), ALGORITHM: AVERAGE

· What is the resulting output of the program below?

· A Priority Queue CONTAINER ADAPTER THAT USES A SPECIFIC UNDERLYING CONTAINER - VECTOR OR DEQUE

· A Priority Queue is an extension of a nnnnn with following properties? QUEUE, AN ELEMENT WITH HIGH PRIORITY IS DEQUEUED BEFORE AN ELEMENT WITH LOW PRIORITY

· Consider a binary nnnnn implemented using an array. Which one of the following array represents a binary nnnnn?

MAX-HEAP, MAX-HEAP - AN ARRAY IN LEVEL ORDER

· Consider a nnnnn, represented by the array: nnnnn. Now consider that a value nnnnn is inserted into this heap. After insertion, the new heap is

MAX-HEAP, ARRAY:40, 30, 20, 10, 15, 16, 17, 8, 4.

VALUE ADDED 35.

NEW HEAP IS 40, 35, 20, 10, 30, 16, 17, 8, 4, 15

· Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the

array (i <= n), the index of the PARENT is FLOOR(I/2)

· Suppose you want to make a priority queue where smaller numbers have the HIGHEST priority. Which of the following would be the best underlying data structure?

MIN-HEAP

· A Huffman tree is a TREE WHERE EACH LEAF REPRESENTS A LETTER

· A Huffman tree for an alphabet of 15? distinct characters has 14? internal nodes and 15 leaf nodes.

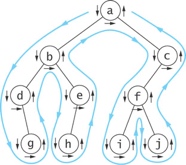
· Should a new Huffman tree be built for each FILE you create?

YES Why or why notDIFFERENT FILES MAY CONTAIN SYMBOLS WITH DIFFERENT FREQUENCIES

OTHER NOTES:

\*A FULL BINARY TREE NEED NOT BE COMPLETE AND A COMPLETE BINARY TREE NEED NOT BE FULL

\*WITH A HEAP, THE LARGEST COMPELTE TREE OF HEIGHT HAS 2^H - 1 NODES.

\*

PRE-ORDER: ABDGEHCFIJ

IN-ORDER: DGBHEAIFJC

POST ORDER:GDHEBIJFCA

A picture containing text, screenshot, font

Description automatically generatedA picture containing text, screenshot, circle, font

Description automatically generatedA picture containing text, screenshot, circle, diagram

Description automatically generated