1. Implement the static method match() in that takes a string s as argument and uses a stack to determine whether its parentheses are properly balanced, and returns true if they are and false otherwise. You may assume that s only consists of parentheses (curly, square, and round).

Example 1:

[()]{}{[()()]()}

Example 2:

[(])

2. Suppose that a minus sign in the input indicates **pop** the stack and write the return value to standard output, and any other string indicates **push** the string onto the stack. Further suppose that following input is processed:

In computer - - science a stack - is an - abstract data - - type -

a. What is written to standard output?

b. What are the contents (top to bottom) left on the stack?

3. Suppose that a minus sign in the input indicates dequeue the queue and write the return value to standard output, and any other string indicates enqueue the string onto the queue. Further suppose that following input is processed:

it was - the - best - of times - - it was - the - - worst - of times -

a. What is written to standard output?

b. What are the contents (head to tail) left on the queue?

4. Suppose that an intermixed sequence of (stack) push and pop operations are performed. The pushes push the integers 0 through 9 in order; the pops print out the return value. Which of the following sequence(s) could not occur?

A 4 3 2 1 0 9 8 7 6 5
B 4 6 8 7 5 3 2 9 0 1
C 2 5 6 7 4 8 9 3 1 0
D 4 3 2 1 0 5 6 7 8 9
E 1 2 3 4 5 6 9 8 7 0
F 0 4 6 5 3 8 1 7 2 9
G 1 4 7 9 8 6 5 3 0 2
H 2 1 4 3 6 5 8 7 9 0

5. Write a method that use recursion to compute the sum of all values in an array. For example, an array arr = { 1, 3, 5, 10} has some of 19.

Scanned with OKEN Scanner

6. Write a recursive method **String reverse(String text)** that reverses a string. For example, reverse("Hello!") returns the string "!olleH". Implement a recursive solution by removing the first character, reversing the remaining text, and combining the two.

7. True/False:

a) Finding the average of N numbers is linear (.   )

b) Suppose we have $f(n) = n^{\lg n}$, $g(n) = 5^n$, then $f(n) = O(g(n))$ (.   )

c) Suppose we have $f(n) = n^{1.4}$, $g(n) = n^{1.2}\lg n^4$, then $f(n) = O(g(n))$ (.   )

8. Below is a table containing a brief description of various data structures that contain a set of integers *S*. For each data structure, think of the best algorithm for solving each of the following two problems:

a) finding the maximum of the collection, which we call *Max(S)*,

b) Deciding if a given integer x is included in the set S, which we call *Member(x, S)*.

Indicate the asymbiotic worst-case running time (using theta notation Θ) of each of these algorithms in the appropriate columns in the table. You should phrase your answer in terms of the number *n* that appears in each description. Unless the description indicates otherwise, you should not make any assumptions about the arrangement of elements in the data structure.

Note:

- You can spell the word theta instead of the symbol Θ for simplicity
- The first row was done as an example

Representation of S	worst-case running time of Max(S)	worst-case running time of Member(x, S)
An ordered array of n integers	theta(n)	Θ (n)
An array of n integers, ordered from largest to smallest	[a]	[b]
An unordered linked list of n integers	[c]	[d]
A binary tree of height n	[e]	[f]

9. Master Method

1. $T(n) = 3T(n/2) + n^2$
2. $T(n) = 4T(n/2) + n^2$
3. $T(n) = T(n/2) + 2^n$
4. $T(n) = 2^nT(n/2) + n^n$
5. $T(n) = 16T(n/4) + n$
6. $T(n) = 2T(n/2) + n \log n$

10. Heaps

11. Insert the following keys in that order into a maximum-oriented heap-ordered binary tree:

E A S Y Q U [illegible] T I O N

a) What is the state of the array **arr** representing the resulting tree?
b) What is the height of the tree (the root is at height *zero*)?

12. Suppose that a letter in the input means *insert the letter* into an initially empty heap and an asterisk (*) means *remove the minimum* from the heap. What is left in the array after the following input is processed?

P R I O * R * * I * T * Y * * * Q U E * * * U E *

13. Show that the worst-case running time of heapsort is Ω (nlgn)

14. Show T(n) =T(n/2) + 1 is O(logn).

15. Show that the solution of T(n) = T(n-1) + n is O($n^{[illegible]}$)

16. Let *f(n)* and *g(n)* be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that max *(f(n), g(n))* = *Θ(f(n) + g(n))*

17. Is $2^{[illegible]}$ = O($2^{[illegible]}$)? Is $2^{[illegible]}$ = O($2^{[illegible]}$)

18. More recurrence examples:

a) **T(n) = 3 T(n/3) + n/lgn**

b) **T(n) = T (n-2) + 1/n**

19. Show that in any sub-tree of a max-heap, the root of the sub-tree contains the largest value occurring **anywhere in that sub-tree.**

20. Is the array with values {32,17, 14, 6, 13 ,10, 1, 5, 7, 12} a max-heap?

21. Hash table

Demonstrate what happens when we insert the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into hash **table with collisions resolved by chaining.** Let the table have 9 slots, and let the hash function be **h(k) = k mod 9.**

22. Consider inserting the following keys into an initially empty hash table of $M = 5$ lists, using separate chaining. Use the hash function $h(k) = k \bmod M$ to transform the kth letter of the alphabet into a table index, where $1 \leq k \leq 26$.

E A S Y Q U T I O N

a. What is the value of h(18)?

23. BST Binary Search trees

For the set of {1,4, 5, 10, 16, 17, 21}, draw binary search trees of height 2, 3, 4, 5, and 6.

24. RBTrees

Red Black Trees

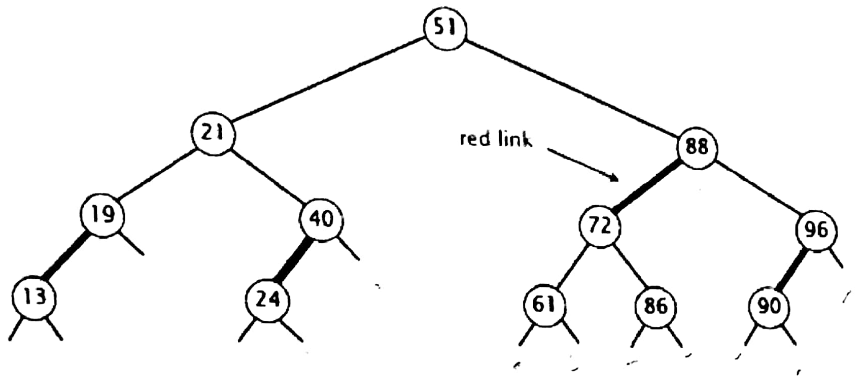
Question1

5 pts

Consider the red-black BST in the figure below. The level order traverse is:

51, 21, 88, 19, 40, 72, 96, 13, 24, 61, 86, 90

If we insert 98 into the red-black BST, what would be the level order traverse of the resulting BST?



move/copy question to another bank

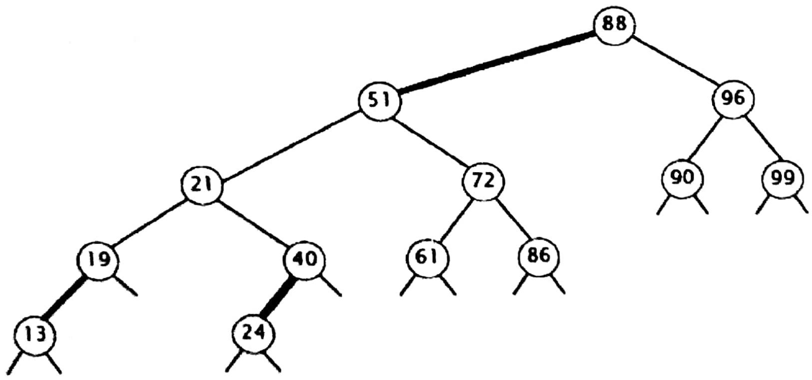
Question2

5 pts

Consider the red-black BST in the figure below. The level order traverse is:

88, 51, 96, 21, 72, 90, 99, 19, 40, 61, 86, 13, 24

If we insert 87 into the red-black BST, what would be the level order traverse of the resulting BST?



25. QuickSort:

1. Illustrate the operation of PARTITION on the array A = {13, 19, 9, 5, 12, 8 ,7 4, 21, 2, 6, 11}

26. Binary search

Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363, which of the following sequences could NOT be the sequence of nodes examined?

a. 2, 252, 401, 398, 330, 344, 397, 363
b. 924, 220, 911, 244, 898, 258, 362, 363
c. 925, 202, 911, 240, 912, 245, 363
d. 2, 399, 387, 219, 266, 382, 381, 278, 363
e. 935, 278, 347, 621, 299, 392, 392, 358, 363

2, 401, 398, 363, 344, 330, 252.