University of Rwanda - College of Science and Technology School of ICT - Computer Science - Year 2

Analysis of Algorithms - CAT 1 October 19, 2022

1. For each of the following pairs of functions, either *f* (*n*) is in *O*(*g*(*n*)), *f* (*n*) is in Ω(*g*(*n*)), or *f* (*n*) = Θ(*g*(*n*)). Determine which relationship is correct and briefly explain why.
   1. *f* (*n*) = *log*(*n*2); *g*(*n*) = *log*(*n*) + 5 [2 marks]

*Answer: f* (*n*) = Θ(*g*(*n*)).

* 1. *f* (*n*) = *√n*; *g*(*n*) = *log*(*n*2) [2 marks]

*Answer: f* (*n*) is in Ω(*g*(*n*)),

1. Place the following functions into increasing asymptotic order. *f*1(*n*) =

*n*2*log*2(*n*), *f*2(*n*) = *n*(*log*2*n*)2, *f*3(*n*) = Σ*n* 2*i*, *f*4(*n*) = *log*2(Σ*n* 2*i*) [4

*i*=0

*i*=0

marks]

Answer: f4->f2->f1->f3

1. A common problem for compilers and text editors is determining whether the parentheses in a string are balanced and properly nested. For example, the string ((())())() contains properly nested pairs of parentheses, which the strings )()( and ()) do not. [4 marks]
   1. Give an algorithm that returns true if a string contains properly nested and balanced parentheses, and false if otherwise. [4 marks]

Answer

1. function checkBracketSequences():
2. print "Please enter sequences of brackets"
3. x = read input from user
4. n = 0
5. for each character c in x:
6. if c is '(':
7. increment n by 1
8. else if c is ')':
9. decrement n by 1
10. print c
11. if n is 0:
12. print "True"
13. else:
14. print "False"

Codes

import jdk.jshell.Snippet;  
  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 System.*out*.println("Please enter sequences of brackets ");  
 Scanner str = new Scanner(System.*in*);  
 String x = str.next();  
// String x = "(()()()()()()()()())()()()(())";  
 int n = 0;  
 char[] gfg = x.toCharArray();  
 for (int i = 0; i < gfg.length; i++) {  
 if(gfg[i] == '('){  
 n++;  
 }else if(gfg[i] == ')'){  
 n--;  
 }  
  
 System.*out*.println(gfg[i]);  
 }  
 if(n == 0){  
 System.*out*.println("very good");  
 }else{  
 System.*out*.println("Very baaad");  
 }  
 }  
}

* 1. What is the time complexity of the algorithm used in (a)? [2 marks]

Answer: O(n)

4. Is the array with values (23; 17; 14; 6; 13; 10; 1; 5; 7; 12) is a max-heap? [ 2 marks]

5. Give an efficient algorithm to rearrange an array of n keys so that all the negative keys precede all the nonnegative keys. Your algorithm must be in-place, meaning you cannot allocate another array to temporarily hold the items. How fast is your algorithm? [5 marks]

1