Data Structures

PSS 1

January 27, 2024

Prologue meme of the day:

[redacted]

Can you guys please recommend books that made you cry?

[redacted]

Data Structures and Algorithms in Java (2nd Edition) 2nd Edition

by Robert Lafore (Author)

114 customer reviews

Kindle	Hardcover	Paperback	Other Se
$29.80	$33.89 - $45.04	$23.39 - $27.18	See all 6 versi

○ Buy used

● Buy new

In Stock.

Asymptotic Analysis

0) Order the following functions by asymptotic growth rate:

$10log(n)$ $5n$ $(n+1)!$ $\binom{n}{n/2}$

$2^{\log\sqrt{n}}$ $n + 5n^3$ 2^{2n} $8^{\log n}$

$5n + n^n$ $\frac{n}{2}$ $nlog(n)$

$n!$ $\binom{n}{3}$ $\log(n!)$

4^n $\log(n^n)$ $\sqrt{n} + 91^{2886}\log(n)$

1) For the following processing times specify their big-Oh complexity

a) $5 + 0.001n^3 + 0.025n$

b) $100n + 0.01n^2$

c) $2n + n^2 + 0.5n^{2.25}$

d) $100n\log_3(n) + n^3 + 100n$

e) $n\log_3(n) + n\log_2(n)$

f) $0.01n\log_2(n) + n(\log_2(n))^2$

2) Specify the big-Oh time complexity for the following methods

a)

```
public static int method1(int[] arr) {
    int length = arr.length;
    int result = 0;
    for(int i = 0; i < length; i++) {
        result += arr[i];
    }
    return result;
}
```

b)

```
public static void bogosort(int[] arr) {
        while (!isSorted(array)) {
            shuffleArray(array);
        }
}

public static boolean isSorted(int[] arr) {
        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] > array[i + 1]) {
                return false;
            }
        }
        return true;
}

public static void shuffleArray(int[] arr) {
        int index, temp;
        Random random = new Random();
        for (int i = array.length - 1; i > 0; i--) {
            index = random.nextInt(i + 1);
            temp = array[index];
            array[index] = array[i];
            array[i] = temp;
        }
}
```

c)

```
Public static void method3(int[] arr) {
    int n = arr.length;
    for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j = j *2) {
             System.out.println(" ");
    }
    }
}
```

d)

```
public static void method4(int n) {
    for (int i = n; i > 0; i /= 2) {
        for (int j = 0; j < n; j++){
            System.out.println(“Something”);
        }
    }
}
```

e)

```
public static void method5(int n) {
    for (int i = n; i > 0; i /= 2) {
        for (int j = 1; j < n; j *= 2) {
            for (int k = 0; k < n; k += 2) {
                System.out.println("Something");
            }
        }
    }
}
```

f)

```
public static int method6(int[] arr) {
    int n = arr.length, total = 0;
    for (int i = 0; i < n; i++)
        arr[i] += 2;
    for (int i = 0; i < n; i++)
        total += arr[i];
    return total;
}
```

Recursion

0) Write a recursive program to calculate GCD of two numbers

Description:

Euclid's Algorithm example:

We must find GCD for a = 1071 and b = 462

First we divide 1071 by 462 and take the remainder (let's denote it r) which is obviously less than 462, then we divide 462 by r and take the remainder… until r = 0

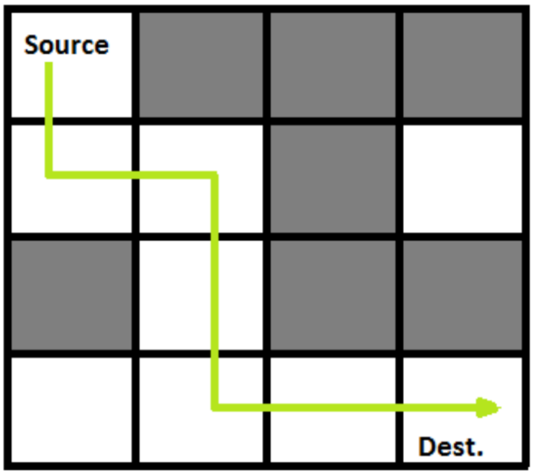
1071/462:	1071 = 2*462 + 147	
462/147:	462 = 3*147 + 21	
147/21:	147 = 7*21 + 0	GCD(1071,462) = 21

```
function gcd(a, b):
find the gcd of b and the remainder r = a/b
for each next call b becomes argument a, and r becomes
argument b
repeat until b becomes 0: if b = 0, then GCD is a
```

1) Write a recursive program to calculate the sum of digits of an integer

Example:
Input: 1850374321
Output: 1 + 8 + 5 + 0 + 3 + 7 + 4 + 3 + 2 + 1, i.e. 34

2) Write a maze solving recursive program which checks whether the destination is reachable from the source. Write a recursive method that receives a 2D array of characters where 'X' represents the grey blocks, '.' Represents the white, 'S' represents the source, and 'D' is the destination.



3) Consider the following function:

Let $x_1, x_2, x_3 \ldots$ be a sequence of real numbers such that:

$$x_i = \begin{cases} 1, \text{ if } i = 0 \\ \left(\sqrt{2}\right)^{x_{i-1}}, \text{ if } i > 0 \end{cases}$$

The initial numbers of this sequence are:

$$x_0 = 1;\ x_1 = \sqrt{2};\ x_2 = \sqrt{2}^{\sqrt{2}};\ x_3 = \sqrt{2}^{\sqrt{2}^{\sqrt{2}}} \ldots$$

Write a recursive program that prints the first 1000 members of this sequence, numbers are represented as double.

Epilogue meme of the day: