Time left 1:18:44

Question 1

Not yet answered

Marked out of 11.00

Fill in the underlined blank lines in the following code.

```
public static  int   roundedLog( int  n) {
    // Pre-Condition:      n is a positive integer.
    // Post-Condition:   returns ⌊log n⌋ , i.e., logarithm of input n base 2 rounded down.
    int r = 0 ;
    while ( n > 1 ) {
          // Loop Invariant: _____          // fill in this line
                _____                    // fill in loop body
    }
   return  r ;
}
```

a. *// Loop Invariant:* the return value stated in the post-condition equals the current value of $r + \lfloor \log n \rfloor$ *// fill in this line*
   n /=2; r++; *// fill in loop body*

b. *// Loop Invariant:* r will eventually become $\lfloor \log n \rfloor$ *// fill in this line*
   n /=2; r++; *// fill in loop body*

c. *// Loop Invariant:* the loop repeatedly increments r and halves n *// fill in this line*
   n /=2; r++; *// fill in loop body*

d. *// Loop Invariant:* the purpose of the loop is to compute $\lfloor \log n \rfloor$ *// fill in this line*
   r = (int) (Math.log10(n) / Math.log10(2)) ; *// fill in loop body*

e. More than one of the other choices are correct.

f. None of the other choices are correct.

Question 2

Not yet answered

Marked out of 8.00

What is type erasure in Java generics?

- a. The process of adding generic type information during compilation
- b. The process of removing all generic type information during compilation
- c. The process of enforcing generic type information at runtime
- d. The process of dynamically updating generic type information during program execution

Question 3

Not yet answered

Marked out of 10.00

The solution to the recurrence relation $T(n) = T(n-2) + n/2$ is $T(n) = \Theta$ (____?____).

- a. n
- b. $n \log n$
- c. None of the other choices.
- d. $\log n$
- e. n^2

Question 4

Not yet answered

Marked out of 11.00

We are given an array $A[0..n-1]$ of n numbers where the first $\log n$ numbers appear in arbitrary order but the remaining $n - \log n$ numbers appear in increasing order. We are also given a search key k and want to find out whether number k appears in array A. The most efficient algorithm to do the search will take Θ (_____?_____) time in the worst case.

- a. n
- b. $\log \log n$
- c. None of the other choices.
- d. $\log n$
- e. $\log^2 n$

Question 5

Not yet answered

Marked out of 10.00

```
/**
 * Pre-Cond:   Input is an integer array A.
 * Post-Cond: Elements of A are rearranged into two groups.
 *            Non-negative elements form the first group;   negative elements the second group.
 *            The relative order of elements within each group remains undisturbed.
 * Example:   [ -4,  6,  2,  8,  -9,  -7,  0,  -5,  3]   becomes   [ 6,  2,  8,  0,  3,  -4,  -9,  -7,  -5].
 */
public static void splitBySign(Integer[] A) {
    // TODO:  place a correct subset of the numbered lines below in correct order here.
}
```

1. int r = A.length -1;
2. r = A.length -1;
3. int r = 0;
4. r = 0;
5. A[r--] = S.pop();
6. A[r++] = S.pop();
7. if (A[i] >= 0) A[r++] = A[i]; else S.push(A[i]);
8. if (A[i] < 0) A[r--] = A[i]; else S.push(A[i]);
9. while (!S.empty())
10. Stack<Integer> S = new Stack<>();
11. for (int i = 0; i < A.length; i++)
12. for (int i = A.length -1; i >= 0; i--)

- a. None of the other choices are correct.
- b. 3, 10, 11, 7, 2, 9, 5.
- c. 3, 10, 11, 7, 9, 5.
- d. 1, 10, 12, 8, 9, 5.
- e. More than one of the other choices are correct.
- f. 10, 3, 12, 7, 9, 5.

Question 6

Not yet answered

Marked out of 8.00

If $T_1(n) = O(f(n))$ and $T_2(n) = O(f(n))$, then $T_1(n) = O(T_2(n))$.

- a. False
- b. True

Question 7

Not yet answered

Marked out of 11.00

The worst-case running time of the algorithm below is $T(n) = \Theta$ (_____?_____).

```
public static double foo( double x , int n) {
    if ( x  < 10 || n < 5 ) return 2*x*(n+5) ;
    double e = 0;
    for ( int i = 0; i < n; i++ )   e += 5*(i+7)*(i+n);
    return foo(x+4 , n/2) + foo(x+7 , n/2) + 5*(e+3)*(n+7) ;
}
```

- a. n
- b. n^2
- c. $n^2 \log n$
- d. This method may not terminate.
- e. $n \log n$
- f. 2^n

Question 8

Not yet answered

Marked out of 11.00

What is the asymptotic running time of the method below as a function of n?

```
public void doubleLoop ( int n ) {
   for ( int k = 1; k < n*n; k +=5 )
      for ( int m = 1; m < k; m *=3  )
          System.out.println( k*m*m );
}
```

- a. $\Theta(n^2 \log n)$
- b. None of the other choices.
- c. $\Theta(n^2)$
- d. $O(n \log n)$
- e. $O(n \log^2 n)$
- f. $\Omega(n^4 \log n)$

Question 9

Not yet answered

Marked out of 10.00

Order the following three functions of n in increasing order of growth rate:

$$F(n) = \frac{3n^2 \log^5 n + 4n^3}{7 \log^2 n + 2\sqrt{6n \log^2 n}}, \quad G(n) = \frac{5n \log^{100} n + 7n^3 \sqrt{n} \log n}{n^{1.7} + n^2 \log n}, \quad H(n) = \frac{5n^4 + n \log n}{n + 5\sqrt{n} \log n}.$$

- a. $F(n) \ll G(n) \ll H(n)$
- b. $H(n) \ll G(n) \ll F(n)$
- c. $G(n) \ll F(n) \ll H(n)$
- d. None of the other choices.
- e. $G(n) \ll H(n) \ll F(n)$
- f. $H(n) \ll F(n) \ll G(n)$

Question 10

Not yet answered

Marked out of 10.00

Consider the following:

1. fully parenthesized infix expression `( ( ( x + 3 ) - ( y * z ) ) / ( ( 7 * y ) - ( 5 * x ) ) )`
2. un-parenthesized postfix expression `5 x * 7 y * - y z * x 3 + - /`
3. un-parenthesized postfix expression `x 3 + y z * - 7 y * 5 x * - /`
4. un-parenthesized prefix expression `/ - + x 3 * y z - * 7 y * 5 x`

The question is which of them are equivalent, i.e., represent exactly the same expression.

- a. No two of them are equivalent.
- b. Only 1, 3 are equivalent.
- c. Only 1, 4 are equivalent.
- d. Only 1, 3, 4 are equivalent.
- e. Only 1, 2, 4 are equivalent.
- f. 1 is equivalent to 3, and 2 is equivalent to 4.

Previous activity
Sample Midterm Tests & Solutions

Jump to...

Next activity
Practice Quiz for Final Exam