DataStructures-Quiz1(S2 AIE B)

Hi, POOJA S PILLAI. When you submit this form, the owner will see your name and email address.

1

(2 Points)

In a competition, four different functions are observed. All the functions use a single for loo for loop, same set of statements are executed. Consider the following for loops:

A) for(i = 0; i < n; i++)

B) for(i = 0; i < n; i += 2)

C) for(i = 1; i < n; i *= 2)

D) for(i = n; i <= n; i /= 2)

If **n** is the size of input(positive), which function is most efficient(if the task to be performed

- (●) A
- ( ) B
- ( ) C
- ( ) D

2

(2 Points)

Which of the following is not O(n^2)?

- ( ) (15) * n^2

- ○ n^1.98
- ◉ n3/^(sqrt(n))
- ○ (20) * n^2

3

(2 Points)

What does it mean when we say that an algorithm X is asymptotically more effi

- ○ X will be a better choice for all inputs except possibly small inputs
- ○ X will be a better choice for all inputs except possibly large inputs
- ○ Y will be a better choice for small inputs
- ◉ X will be a better choice for all input

4

(2 Points)

```
int fun1 (int n)
{
  int i, j, k, p, q = 0;
  for (i = 1; i < n; ++i)
  {
    p = 0;
    for (j = n; j > 1; j = j/2)
      ++p;
    for (k = 1; k < p; k = k*2)
      ++q;
  }
  return q;
}
```

Find Time complexity?

- ○ n3
- ○ n (logn)^2
- ○ nlogn

- [x] nlog(logn)

5

(1 Point)

Consider a two dimensional array A[20][10]. Assume 4 words per memory cell, the base a
100, elements are stored in row-major order and first element is A[0][0]. What is the addr

- [ ] 560
- [x] 460
- [ ] 570
- [ ] 575

6

(2 Points)

Find the time complexity?

```
int fun(int n)
{
    int count = 0;
    for (int i = 0; i < n; i++)
        for (int j = i; j > 0; j--)
            count = count + 1;
    return count;
}
```

- [x] O(n^2)
- [ ] O(n*log(n))
- [ ] O(n)
- [ ] O(n*log(n*Log(n)))

7

(1 Point)

Let w(n) and A(n) denote respectively, the worst case and average case running time of an a on an input of size n. which of the following is ALWAYS TRUE?

- A(n) = Omega(W(n))
- A(n) = Theta(W(n))
- A(n) = O(W(n))
- A(n) = o(W(n))

8

(1 Point)

```
double foo (int n){
   int i;
   double sum;
   if (n = = 0) return 1.0;
   else{
      sum = 0.0;
      for (i = 0; i < n; i++)
         sum += foo (i);
      return sum;
   }
}
```

What is the space complexity?

- O(1)
- O(n)
- O(n!)
- O(n^n)

9

(2 Points)

Consider the following functions:
f(n) = 2n
g(n) = n!
h(n) = nlog(n)
Which of the following statements about the asymptotic behavior of f(n), g(n), a

- ○ f(n) = O(g(n)); g(n) = O(h(n))
- ○ f(n) = Omega (g(n)); g(n) = O(h(n))
- ● g(n) = O(f(n)); h(n) = O(f(n))
- ○ h(n) = O(f(n)); g(n) = Omega (f(n))

10

(1 Point)

Suppose you are given an array s[1..n] and a procedure reverse (s, i, j) which reverses the o
a between positions i and j (both inclusive). What does the following sequence do, where
reverse(s, 1, k) ;
reverse(s, k + 1, n);
reverse(s, l, n);

- ● Rotates s left by k positions
- ○ Leaves s unchanged
- ○ Reverses all elements of s
- ○ None of the above

11

(2 Points)

Let A be a two dimensional array declared as follows:
A: array [1 ... 10] [1 ... 15] of integer;
Assuming that each integer takes one memory location, the array is stored in row-major
element of the array is stored at location 100, what is the address of the element a[i][j] ?

- (●) 15i+ j+ 84
- ( ) 15j+ i+ 84
- ( ) 10i+ j+ 89
- ( ) 10j+ i+ 89

12

(2 Points)

Find out the time complexity?

```
int fun(int n)
{
    int count = 0;
    for (int i = n; i > 0; i /= 2)
        for (int j = 0; j < i; j++)
            count += 1;
    return count;
}
```

- ( ) O(n^2)
- (●) O(n*log(n))
- ( ) O(n)
- ( ) O(n*log(n*Log(n)))

13

(2 Points)

A three dimensional array is declared as int A[x][y][z]. Consider that array elements are sto order and indexing begins from 0. Here, the address of an item at the location A[p][q][r] c follows (where w is the word length of an integer):

- ○ &A[0][0][0] + w(y * z * p + z*q + r)
- ● &A[0][0][0] + w(x * y * p + z * q+ r)
- ○ &A[0][0][0] + w(x * y * q + z * p + r)
- ○ &A[0][0][0] + w(y * z * q + z * p + r)

14

(1 Point)

Consider the following function,

```
int fun(int n)
{
 int i, j, k = 0;
 for (i = n/2; i <= n; i++)
 for (j = 2; j <= n; j = j * 2)
 k = k + n/2; return k;
 }
```

What is the time complexity of the function?

- ○ n^2
- ● n logn
- ○ n^3
- ○ n^3logn

15

(1 Point)

A program P reads in 500 integers in the range [0..100] representing the scores of 500 stu the frequency of each score above 50. What would be the best way for P to store the freq

- ● An array of 50 numbers
- ○ An array of 100 numbers

- ◯ An array of 500 numbers
- ◯ A dynamically allocated array of 550 numbers

16

(1 Point)

Which of the given options provides the increasing order of asymptotic complexity of fun f4?
f1(n) = 2^n f2(n) = n^(3/2) f3(n) = n*log(n) f4(n) = n^log(n)

- ◉ f3, f2, f4, f1
- ◯ f3, f2, f1, f4
- ◯ f2, f3, f1, f4
- ◯ f2,f3,f4,f1

⚠ This form is currently not accepting responses.