

الوقت المتبقي: 59 دقائق, 43 ثانية (ثوان).

حالة إكمال الأسئلة:

السؤال 1

.Find the recurrence relation of the below code

```
} function1 (long x, long n)
    if(n == 0)
        ;return 1
    else if(n == 1)
        ;return x
    else if ((n % 2) == 0)
        ;return power (x, n/2) * power (x, n/2)
    else
        ;return x * power (x, n/2) * power (x, n/2)
    {
```

$T(n) = c$, if $n = 0$ or $n = 1$ and $T(n) = 2T(n/2) + b$ if $n > 2$

$T(n) = 2T(n/2) + b$ if $n > 2$

$T(n) = c$, if $n = 0$ or $n = 1$

None of the above

اكتب هنا للبحث

فرض الإكمال يمكن حفظ هذا الاختبار واستئنافه عند أية نقطة حتى لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات على

الوقت المتبقي: 56 دقائق. 37 ثانية (ثوانٍ).

حالة إكمال الأسئلة:

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 2

?Which one is not the feature of an algorithm

Effectiveness .1

Definiteness .2

Complexity .3

Finiteness .4

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

اكتب هنا للبحث



الوقت المتبقي: 54 دقائق. 49 ثانية (نواب).

حالة إكمال الأسئلة:

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 3

? What is Base Case of the recurrence relation

- .The portion of the definition that does not contain T
- .The portion of the definition that does contain T
- .The complete portion of the definition
- .All of the above

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



N



اكتب هنا للبحث

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 4

Algorithm is randomly defined computational procedure that takes arbitrary value as input and produces arbitrary values as output

صواب

خطأ

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



اكتب هنا للبحث

DELL

الوقت المتبقي: 51 دقائق, 59 ثانية (ثوان).

حالة إكمال الأسئلة:

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 5

.Why we need algorithm analysis? This question has more than one correct answer
If the program is run on a small data set, then the running time becomes an issue

.1

!The program may be efficient .2

writing a working program is not good enough

.3

!The program may be inefficient .4

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

⏪ ⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 8

:Determine its big-O complexity

```
long factorial (int n)
    if (n == 0) }
        ;return 1
    else
        ;return n * factorial (n - 1)
    {
```

$O(n^2)$

$O(n)$

$O \log(n)$

$O(n^3)$



اكتب هنا للبحث 🔍



محاولة متعددة غير مسموح. يمكن إجراء هذا الاختبار مرة واحدة فقط.

فرض الإكمال يمكن حفظ هذا الاختبار واستئنافه عند أية نقطة حتى ينتهي الوقت. سيستمر المؤقت في التشغيل إذا

لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات على الإجابة بعد التقديم.

الوقت المتبقي: 39 دقائق، 14 ثانية (ثوان).

حالة إكمال الأسئلة:

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 9

A _____ is a set of instructions which the computer will follow to solve a problem

Program

Function

Procedure

Process

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

اكتب هنا للبحث



الوقت المتبقي: 35 دقائق. 38 ثانية (نوان).

حالة إكمال الأسئلة:

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 11

Find the running time complexity of the below code

```
public static void hanoi(int n, char from, char to, char temp)
    if (n == 1)
        System.out.println(from + " -----> " + to)
    }else
        hanoi(n - 1, from, temp, to)
        System.out.println(from + " -----> " + to)
        hanoi(n - 1, temp, to, from)
```

- $O(n)$
- $O(k^n)$
- $O(3^n)$
- $O(2^n)$

لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات على الإجابة بعد التقديم.

الوقت المتبقي: 32 دقائق، 18 ثانية (نواب).

حالة إكمال الأسئلة:

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

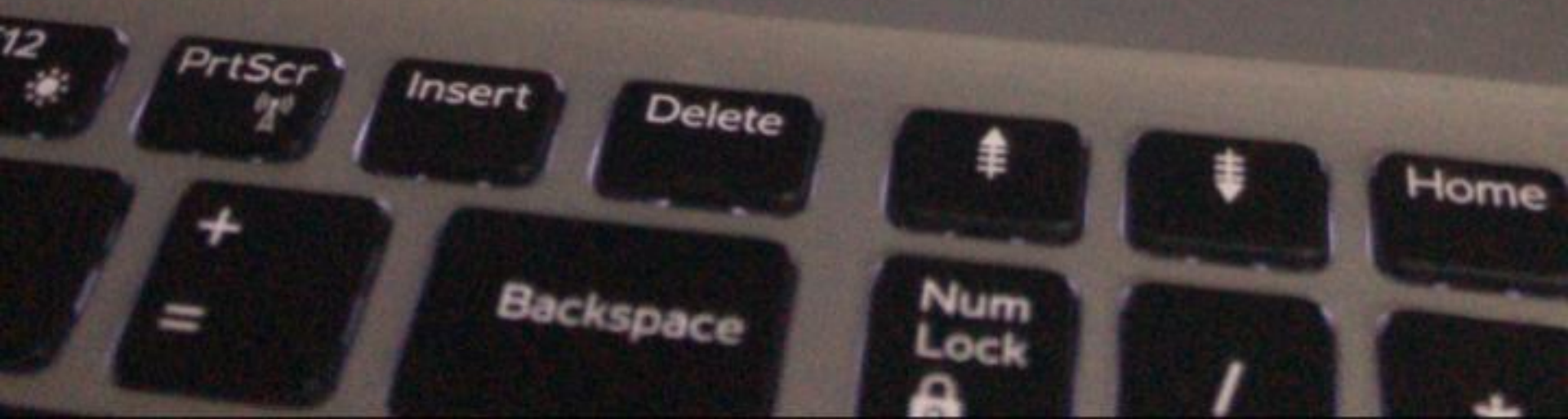
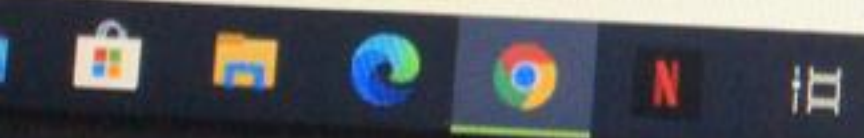
سؤال 12

An algorithm is based largely on optimism and often with minimal theoretical support is called

- 1. Heuristic
- 2. Probabilistic
- 3. Deterministic
- 4. None of the above

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

اكتب هنا للبحث



اختبار محدد بوقت هذا الاختبار له حد زمني يصل إلى 1 ساعة/ساعات. سيتم حفظ هذا الاختبار وإرساله تلقائياً عند
تظهر التحذيرات عند مرور نصف الوقت، وهو 5 دقائق، ويصبح المتبقي من الوقت دقيقاً
محاولات متعددة غير مسموح. يمكن إجراء هذا الاختبار مرة واحدة فقط.
فرض الإكمال يمكن حفظ هذا الاختبار واستئنافه عند أية نقطة حتى ينتهي الوقت. سيستمر المؤقت في
لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات على الإجابة بعد التقديم.

الوقت المتبقي: 30 دقائق، 20 ثانية (ثوان).

حالة إكمال الأسئلة:

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 13

Algorithm can be said as a tool for solving a well specified computational problem

صواب

خطأ

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



اكتب هنا للبحث

محاولات متعددة غير مسموح. يمكن إجراء هذا الاختبار مرة واحدة فقط.

فرض الإكمال يمكن حفظ هذا الاختبار واستئنافه عند أية نقطة حتى ينتهي الوقت. سيستمر الم

لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات على الإجابة بعد التقديم.

بأقل من نصف الوقت. الوقت المتبقي: 25 دقائق. 03 ثانية (ثوان).

حالة إكمال الأسئلة:

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 15

.In this algorithm, the probability of chosen each value is known and controlled

- 1. Randomized
- 2. Probabilistic
- 3. Deterministic
- 4. Non-deterministic

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



اكتب هنا للبحث



بأقل من نصف الوقت. الوقت المتبقي: 22 دقائق. 20 ثانية (توان).

حالة إكمال الأسئلة:

السؤال 16

:Write the recurrence relation for the following method

```

} public void f (int n)
    } if (n > 0)
;System.out.println(n)
    ;f(n-1)
    {
    {

```

$T(0) = a$ for some constant a
 $T(n) = b + T(n - 1)$ for some constant b

$T(0) = a$ for some constant a
 $T(n) = T(n - 1)$ for some constant b

$T(0) = a$ for some constant a
 $T(n) = T(n - 1) + 1$ for some constant b

None of the above



اكتب هنا للبحث



بأقل من نصف الوقت. الوقت المتبقي: 20 دقائق. 05 ثانية (لوان).

حالة إكمال الأسئلة:

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 17

Find the big-Oh running time of the following recurrence. Use the Master Theorem

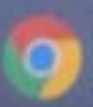
$O(n \log n^2)$

$O(n)$

$O(\log n)$

$O(n \log n)$

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



اكتب هنا للبحث

⚠️ ⬅️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 18

?Which one is the correct definition of the algorithm

a sequence of computational steps that transform the output into the input

is any undefined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output

a collection of any number of lines and codes

is a tool for solving a well-specified computational problem

⚠️ ⬅️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

You have to match from one column to another column

$$2^n$$

$$\text{Log } N$$

$$N^2$$

$$N$$

$$C$$

$$K$$

$$e^x$$

$$\log^2 N$$

$$N \text{ Log } N$$

$$N^0$$

- 1. Quadratic
- 2. Exponential
- 3. Linear
- 4. Constant
- 5. Logarithmic
- 6. Log squared
- 7. Constant
- 8. Constant
- 9. Logarithmic
- 10. Exponential

محاولة متعددة غير مسموح. يمكن إجراء هذا الاختبار مرة واحدة فقط.
فرض الإكمال يمكن حفظ هذا الاختبار واستئنافه عند أية نقطة.
لا يسمح هذا الاختبار بالرجوع. يحظر إجراء تغييرات.

بأقل من نصف الوقت. الوقت المتبقي: 06 دقائق. 59

حالة إكمال الأسئلة:

انقر فوق إرسال لإكمال هذا التقييم.

السؤال 20

? What do you analyze in algorithm

- Correctness
- Amount of work
- Amount of space
- Hard disk

انقر فوق إرسال لإكمال هذا التقييم.

اكتب هنا للبحث



الوقت المتبقي: 45 دقائق, 47 ثانية (ثوان).

حالة إكمال الأسئلة:

$$\begin{aligned} T(n) &= c && \text{if } n = 3 \text{ or } n = 4 \\ T(n) &= T(n-1) + T(n-2) + b && \text{if } n < 2 \end{aligned} \quad \text{○}$$

$$\begin{aligned} T(n) &= c && \text{if } n = 1 \text{ or } n = 2 \\ T(n) &= T(n-2) + b && \text{if } n > 2 \end{aligned} \quad \text{○}$$

$$\begin{aligned} T(n) &= c && \text{if } n = 1 \text{ or } n = 2 \\ T(n) &= T(n-2) + T(n-2) + b && \text{if } n > 2 \end{aligned} \quad \text{○}$$

$$\begin{aligned} T(n) &= c && \text{if } n = 1 \text{ or } n = 2 \\ T(n) &= T(n-1) + T(n-2) + b && \text{if } n > 2 \end{aligned} \quad \text{○}$$

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.



اكتب هنا للبحث



الوقت المتبقي: 37 دقائق. 49 ثانية (توان).

حالة إكمال الأسئلة:

السؤال 20

What is the complexity of the below code

```
public int Search (int target, int[] array  
    (int low, int high  
        if (low > high)  
            ;return -1  
        } else  
            ;int middle = (low + high)/2  
            if (array[middle] == target)  
                ;return middle  
            else if(array[middle] < target)  
                ;return binarySearch(target, array, middle + 1, high)  
            else  
                ;return binarySearch(target, array, low, middle - 1)  
            {  
                {
```

- $O(\log n)$
- $O(n)$
- $O(\text{Long } n/2)$
- None of the above

بأقل من نصف الوقت. الوقت المتبقي: 28 دقائق. 07 ثانية (لوان).

حالة إكمال الأسئلة:

{
{

$T(1) = a$ if $n = 1$ (one element array)
 $T(n) = T(n / 2)$ if $n > 1$

$T(1) = a$ if $n = 1$ (one element array)
 $T(n) = T(n / 2) + b$ if $n > 1$

$T(1) = a$ if $n = 0$ (one element array)
 $T(n) = T(n / 2) + b$ if $n > 1$

None of the above

⚠️ يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

الوقت المتبقي: 45 دقائق. 56 ثانية (ثوان).

حالة إكمال الأسئلة:

:Find the recurrence relation of the below code

```
long fibonacci (int n) { // Recursively calculates Fibonacci number
```

```
    if( n == 1 || n == 2)
```

```
        ;return 1
```

```
    else
```

```
        ;return fibonacci(n - 1) + fibonacci(n - 2)
```

```
}
```

$T(n) = c$ if $n = 3$ or $n = 4$

$T(n) = T(n - 1) + T(n - 2) + b$ if $n < 2$

$T(n) = c$ if $n = 1$ or $n = 2$

$T(n) = T(n - 2) + b$ if $n > 2$

$T(n) = c$ if $n = 1$ or $n = 2$

$T(n) = T(n - 2) + T(n - 2) + b$ if $n > 2$

باقي أقل من نصف الوقت. الوقت المتبقي: 28 دقائق. 16 ثانية (أوان).

حالة إكمال الأسئلة:

```

public int F1 (int target, int[] array
    (int low, int high
        if (low > high)
            ;return -1
        } else
            ;int middle = (low + high)/2
            if (array[middle] == target)
                ;return middle
            else if(array[middle] < target)
                ;return binarySearch(target, array, middle + 1, high)
            else
                ;return binarySearch(target, array, low, middle - 1)
    }
}
    
```

$T(1) = a$ if $n = 1$ (one element array)
 $T(n) = T(n / 2)$ if $n > 1$

$T(1) = a$ if $n = 1$ (one element array)
 $T(n) = T(n / 2) + b$ if $n > 1$