

Comenzado el viernes, 19 de noviembre de 2021, 11:55

Estado Finalizado

Finalizado en viernes, 19 de noviembre de 2021, 12:29

Tiempo empleado 34 minutos 9 segundos

Puntos 18/20

Calificación 9 de 10 (90%)

Pregunta **1**

Correcta

Puntúa 1 sobre 1

¿Cuál es la diferencia entre el *peor caso* y el *caso promedio* en el análisis de algoritmos?

Seleccione una:

- ☐ a. El *peor caso* es la configuración de datos de entrada más desfavorable para el algoritmo, mientras que el *caso promedio* describe una configuración de datos pensada para favorecer al algoritmo.
- ☐ b. El *peor caso* es la configuración de datos de entrada más favorable para el algoritmo, mientras que el *caso promedio* describe una configuración de datos pensada para desfavorecer al algoritmo.
- ☐ c. El *peor caso* es la configuración de datos de entrada más favorable para el algoritmo, mientras que el *caso promedio* describe una configuración aleatoria de datos (no pensada ni para favorecer ni para desfavorecer al algoritmo)
- ☒ d. El *peor caso* es la configuración de datos de entrada más desfavorable para el algoritmo, mientras que el *caso promedio* describe una configuración aleatoria de datos (no pensada ni para favorecer ni para desfavorecer al algoritmo). ✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

El *peor caso* es la configuración de datos de entrada más desfavorable para el algoritmo, mientras que el *caso promedio* describe una configuración aleatoria de datos (no pensada ni para favorecer ni para desfavorecer al algoritmo).

Pregunta **2**

Correcta

Puntúa 1 sobre 1

Suponga que dispone de cuatro algoritmos diferentes para resolver el mismo problema, y que se sabe que los tiempos de ejecución (en el peor caso) son, respectivamente: **$O(n \cdot \log(n))$** , **$O(n^2)$** , **$O(n^3)$** y **$O(n)$** .

¿Cuál de esos tres algoritmos debería elegir, suponiendo que todos hacen el mismo consumo razonable de memoria?

Seleccione una:

- ☐ a. El algoritmo cuyo tiempo de ejecución es $O(n \cdot \log(n))$
- ☒ b. El algoritmo cuyo tiempo de ejecución es $O(n)$
- ☐ c. El algoritmo cuyo tiempo de ejecución es $O(n^2)$
- ☐ d. El algoritmo cuyo tiempo de ejecución es $O(n^3)$

✓ ¡Correcto! Efectivamente, este algoritmo sería el más rápido...

¡Correcto!

La respuesta correcta es:

El algoritmo cuyo tiempo de ejecución es $O(n)$

Pregunta **3**

Correcta

Puntúa 1 sobre 1

¿Qué significa decir que un algoritmo dado tiene un tiempo de ejecución **$O(\log(n))$** ?

Seleccione una:

- ☐ a. El tiempo de ejecución es constante, sin importar la cantidad de datos.
- ☒ b. A medida que aumenta el número de datos, aumenta el tiempo pero en forma muy suave: el conjunto de datos se divide en dos. se procesa una de las mitades, se desecha la otra y se repite el proceso hasta que no pueda volver a dividirse la mitad que haya quedado. ✓ ¡Correcto!
- ☐ c. El proceso normalmente consiste en dos ciclos (uno dentro del otro) de aproximadamente n iteraciones cada uno, de forma que las operaciones críticas se aplican un número cuadrático de veces.
- ☐ d. El tiempo de ejecución es lineal: si aumenta el número de datos, aumenta el tiempo en la misma proporción.

¡Correcto!

La respuesta correcta es:

A medida que aumenta el número de datos, aumenta el tiempo pero en forma muy suave: el conjunto de datos se divide en dos. se procesa una de las mitades, se desecha la otra y se repite el proceso hasta que no pueda volver a dividirse la mitad que haya quedado.

Pregunta 4

Correcta

Puntúa 1 sobre 1

¿Cuáles de las siguientes son características **correctas** del algoritmo *Shellsort*? (Más de una puede ser cierta... marque TODAS las que considere válidas)

Seleccione una o más de una:

- ☒ a. El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Inserción Directa* (o *Inserción Simple*), consistente en armar suconjuntos ordenados con elementos a distancia $h > 1$ en las primeras fases, y terminar con $h = 1$ en la última. ✓ ¡Correcto!
- ☐ b. El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Selección Directa*, consistente en buscar iterativamente el menor (o el mayor) entre los elementos que quedan en el vector, para llevarlo a su posición correcta, pero de forma que la búsqueda del menor en cada vuelta se haga en tiempo logarítmico.
- ☐ c. En el caso promedio, el algoritmo *Shellsort* es tan eficiente como el *Heapsort* o el *Quicksort*, con tiempo de ejecución $O(n \log(n))$.
- ☒ d. El algoritmo *Shellsort* es complejo de analizar para determinar su rendimientos en forma matemática. Se sabe que para la serie de incrementos decrecientes usada en la implementación vista en las clases de la asignatura, tiene un tiempo de ejecución para el peor caso de $O(n^{1.5})$. ✓ ¡Correcto!

¡Correcto!

Las respuestas correctas son:

El algoritmo *Shellsort* consiste en una mejora del algoritmo de *Inserción Directa* (o *Inserción Simple*), consistente en armar suconjuntos ordenados con elementos a distancia $h > 1$ en las primeras fases, y terminar con $h = 1$ en la última.,

El algoritmo *Shellsort* es complejo de analizar para determinar su rendimientos en forma matemática. Se sabe que para la serie de incrementos decrecientes usada en la implementación vista en las clases de la asignatura, tiene un tiempo de ejecución para el peor caso de $O(n^{1.5})$.

Pregunta 5

Correcta

Puntúa 1 sobre 1

Se tiene un algoritmo que realiza cierta cantidad de procesos sobre un conjunto de n datos y un minucioso análisis matemático ha determinado que la cantidad de procesos que el algoritmo realiza en el peor caso viene descripto por la función $f(n) = 3n^3 + 5n^2 + 2n^{1.5}$
¿Cuál de las siguientes expresiones representa mejor el orden del algoritmo para el peor caso?

Seleccione una:

- ☐ a. $O(3n^3 + 5n^2 + 2n^{1.5})$
- ☐ b. $O(n^{1.5})$
- ☐ c. $O(n^3 + n^2)$
- ☒ d. $O(n^3)$

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

$O(n^3)$

Pregunta **6**

Correcta

Puntúa 1 sobre 1

¿Cuál es la **principal** característica de todos los métodos de ordenamiento conocidos como **métodos simples** o **directos**?

Seleccione una:

- ☐ a. Son muy fáciles de programar.
- ☐ b. Son muy veloces para cualquier tamaño del arreglo a ordenar.
- ☒ c. Tienen un tiempo de ejecución de orden cuadrático.
- ☐ d. Tienen un tiempo de ejecución de orden lineal.

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

Tienen un tiempo de ejecución de orden cuadrático.

Pregunta **7**

Correcta

Puntúa 1 sobre 1

¿Qué elementos son necesarios para que una **función recursiva** se considere **bien planteada**?

Seleccione una:

- ☐ a. La función debe tener al menos un ciclo en su bloque de acciones, y ese ciclo debe estar planteado de forma tal que nunca entre en un lazo infinito.
- ☐ b. La función debe tener al menos una invocación a si misma en su bloque de acciones, y después de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema.
- ☒ c. La función debe tener al menos una invocación a si misma en su bloque de acciones, y antes de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema.
- ☐ d. La función deben tener al menos una invocación a si misma en su bloque de acciones.

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

La función debe tener al menos una invocación a si misma en su bloque de acciones, y antes de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema.

Pregunta **8**

Correcta

Puntúa 1 sobre 1

¿Qué hace el siguiente programa (que incluye una función recursiva)?

```
def repetir(numero, rep, res):  
    if rep != 0:  
        res.append(numero)  
        repetir(numero, (rep - 1), res)  
  
v = [1, 3, 3, 7]  
res = []  
for i in range(len(v)):  
    repetir(v[i], 2, res)  
print(res)
```

Seleccione una:

- ☒ a. Genera y muestra un nuevo vector res con cada elemento del vector original v repetido dos veces.
- ☐ b. Genera y muestra un nuevo vector res con los números que en el vector original v eran negativos.
- ☐ c. Produce un error de ejecución al intentar el print() final, ya que el vector res no es generado en ningún momento.
- ☐ d. Produce un error de ejecución, porque la función recursiva no tiene valor de retorno.

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

Genera y muestra un nuevo vector res con cada elemento del vector original v repetido dos veces.

Pregunta 9

Correcta

Puntúa 1 sobre 1

El cálculo del valor a^n (para simplificar, asumimos $a > 0$ y $n \geq 0$ y sabiendo que si $n = 0$ entonces $a^0 = 1$) es igual a multiplicar n veces el número a por sí mismo. Por caso, $5^3 = 5 * 5 * 5 = 125$. Sabiendo esto, ¿cuál de las siguientes sería una *definición recursiva* matemáticamente correcta de la operación $potencia(a, n)$?

Seleccione una:

✓ ¡Correcto!

☒ a.

$$potencia(a, n) \begin{cases} = 1 & \text{si } n == 0 \\ = a * potencia(a, n-1) & \text{si } n > 0 \end{cases}$$

(a y n enteros, a > 0 y n >= 0)

☐ b.

$$potencia(a, n) \begin{cases} = 1 & \text{si } n == 0 \\ = a * potencia(a, n) & \text{si } n > 0 \end{cases}$$

(a y n enteros, a > 0 y n >= 0)

☐ c.

$$potencia(a, n) \begin{cases} = 1 & \text{si } n == 0 \\ = a * potencia(a-1, n) & \text{si } n > 0 \end{cases}$$

(a y n enteros, a > 0 y n >= 0)

☐ d.

$$potencia(a, n) \begin{cases} = 1 & \text{si } n == 0 \\ = a + potencia(a, n-1) & \text{si } n > 0 \end{cases}$$

(a y n enteros, a > 0 y n >= 0)

¡Correcto!

La respuesta correcta es:

$$potencia(a, n) \begin{cases} = 1 & \text{si } n == 0 \\ = a * potencia(a, n-1) & \text{si } n > 0 \end{cases}$$

(a y n enteros, a > 0 y n >= 0)

¿Cuáles de las siguientes afirmaciones **son ciertas** en relación a conceptos asociados con la **recursividad**?

Seleccione una o más de una:

- ☐ a. Una función recursiva no puede incluir más de una invocación a sí misma en su bloque de acciones.
- ☒ b. Cada instancia recursiva que es ejecutada, almacena dos grupos de datos en el stack segment: la dirección de retorno (a la que se debe regresar cuando termine la ejecución de esa instancia) y las variables locales que esa instancia de la función haya creado. ✓ ¡Correcto! De hecho, estos datos son almacenados en el stack segment por toda función que sea invocada, haya o no recursividad implicada.
- ☒ c. A medida que se desarrolla la cascada de invocaciones recursivas, el stack segment se va llenando para darle soporte a cada instancia recursiva, y luego, cuando una instancia logra finalizar y se produce el proceso de vuelta atrás, el stack segment comienza a vaciarse. ✓ ¡Correcto!
- ☒ d. Si una función es recursiva, entonces no debe incluir ningún ciclo en su bloque de acciones. ✗ Incorrecto... una función recursiva puede contener lo que sea que el programador necesite. Un ejemplo: revise el algoritmo Quicksort o el programa para las Ocho Reinas...

Las respuestas correctas son:

Cada instancia recursiva que es ejecutada, almacena dos grupos de datos en el stack segment: la dirección de retorno (a la que se debe regresar cuando termine la ejecución de esa instancia) y las variables locales que esa instancia de la función haya creado.,

A medida que se desarrolla la cascada de invocaciones recursivas, el stack segment se va llenando para darle soporte a cada instancia recursiva, y luego, cuando una instancia logra finalizar y se produce el proceso de vuelta atrás, el stack segment comienza a vaciarse.

Pregunta **11**

Incorrecta

Puntúa 0 sobre 1

Considere la siguiente función (vista en clases) basada en **backtracking** para resolución del **Problema de las Ocho Reinas**, e indique cuál de las opciones que se muestran es correcta:

```
def intend(col):  
    global rc, qr, qid, qnd  
  
    fil, res = -1, False  
    while not res and fil != 7:  
        res = False  
        fil += 1  
        di = col + fil  
        dn = (col - fil) + 7  
        if qr[fil] and qid[di] and qnd[dn]:  
            rc[col] = fil  
            qr[fil] = qid[di] = qnd[dn] = False  
  
            if col < 7:  
                res = intend(col + 1)  
                if not res:  
                    qr[fil] = qid[di] = qnd[dn] = True  
                    rc[col] = -1  
            else:  
                res = True  
  
    return res
```

Seleccione una:

- ☐ a. La función es correcta y funciona sin problemas.
- ☐ b. La función no es correcta porque la diagonal normal debe almacenar valores (**columna + fila**).
- ☒ c. La función no es correcta porque **qr** debe señalar las columnas disponibles y no **✗** **Incorrecto...** Los valores de **qr** efectivamente deben señalar filas...
- ☐ d. La función no es correcta porque no debe asignarse a **rc[col]** el valor de -1.

Revise el funcionamiento y la explicación de este algoritmo en la Ficha 28.

La respuesta correcta es:

La función es correcta y funciona sin problemas.

Pregunta **12**

Correcta

Puntúa 1 sobre 1

¿Cuál es la cantidad de niveles del *árbol de invocaciones recursivas* que se genera al ejecutar el *Quicksort* para ordenar un arreglo de n elementos, en el **peor caso**? (Es decir: ¿Cuál es la *altura* de ese árbol en ese peor caso?)

Seleccione una:

- ☐ a. Altura = $\log(n)$
- ☐ b. Altura = $n * \log(n)$
- ☐ c. Altura = n^2
- ☒ d. Altura = n

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

Altura = n

Pregunta **13**

Correcta

Puntúa 1 sobre 1

Dada la siguiente función recursiva del factorial:

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n - 1)
```

Seleccione cuál de las siguientes opciones es la correcta, si se le pasa como parámetro actual a esta función el valor -4.

Seleccione una:

- ☐ a. La función devuelve como resultado: 24.
- ☐ b. La función devuelve como resultado: -24.
- ☒ c. Así como está planteada, si entra el -4 (o cualquier otro negativo) como parámetro, la condición de corte nunca es cierta y se producirá un error de stack overflow en algún momento ✓ ¡Correcto! La condición de corte solo prevé la aparición del cero, pero si el parámetro es negativo nunca se llegará al cero restando uno a ese valor...
- ☐ d. La función no tiene definido ningún caso base, que es aquel caso que se resuelve sin recursividad, y por lo tanto no está bien planteada.

¡Correcto!

La respuesta correcta es:

Así como está planteada, si entra el -4 (o cualquier otro negativo) como parámetro, la condición de corte nunca es cierta y se producirá un error de stack overflow en algún momento

Pregunta **14**

Correcta

Puntúa 1 sobre 1

¿Cuál de los siguientes esquemas simples de pseudocódigo representa a un algoritmo planteado mediante estrategia Divide y Vencerás, pero con tiempo de ejecución $O(n \cdot \log(n))$?

Seleccione una:

- ☐ a. `proceso(partición):`
 `adicional() [==> $O(n^3)$]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☒ b. `proceso(partición):`
 `adicional() [==> $O(n)$]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☐ c. `proceso(partición):`
 `adicional() [==> $O(n^2)$]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☐ d. `proceso(partición):`
 `adicional() [==> $O(1)$]`
 `proceso(partición/2)`
 `proceso(partición/2)`

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

```
proceso(partición):  
    adicional() [==>  $O(n)$ ]  
    proceso(partición/2)  
    proceso(partición/2)
```

Pregunta **15**

Correcta

Puntúa 1 sobre 1

Considere el problema de las *Ocho Reinas* presentado en clases. Se ha indicado que se puede usar un arreglo rc de componentes, en el cual el casillero $rc[col] = fil$ indica que la reina de la columna col está ubicada en la fila fil . ¿Cuáles de las siguientes configuraciones para el arreglo rc representan **soluciones incorrectas** para el problema de las *Ocho Reinas*? (Más de una respuesta puede ser cierta, por lo que marque todas las que considere correctas...)

Seleccione una o más de una:

☐ a. $rc = [4, 7, 3, 0, 2, 5, 1, 6]$

☐ b. $rc = [5, 3, 6, 0, 7, 1, 4, 2]$

☒ c. $rc = [3, 5, 7, 0, 5, 1, 2, 4]$

☒ d. $rc = [2, 0, 7, 4, 5, 1, 6, 3]$

✓ ¡Correcto! Efectivamente, si rc contiene los valores mostrados, hay al menos un problema: las reinas de las columnas 1 y 4 están ubicadas en la misma fila: la 5.

✓ ¡Correcto! Efectivamente, si rc contiene los valores mostrados, **hay al menos** un problema: las reinas de las columnas 3 y 4 están ubicadas en la misma diagonal normal: la diagonal -1.

¡Correcto!

Las respuestas correctas son:

$rc = [3, 5, 7, 0, 5, 1, 2, 4]$,

$rc = [2, 0, 7, 4, 5, 1, 6, 3]$

Pregunta **16**


Parcialmente correcta

Puntúa 1 sobre 1

¿Cuáles de las siguientes afirmaciones son correctas respecto del algoritmo de backtracking que se propuso en clases para el problema de las Ocho Reinas (que mostramos más abajo)? (Más de una respuesta puede ser cierta, por lo que marque todas las que considere correctas).

```
def intend(col):  
    global rc, qr, qid, qnd  
  
    fil, res = -1, False  
    while not res and fil != 7:  
        res = False  
        fil += 1  
        di = col + fil  
        dn = (col - fil) + 7  
        if qr[fil] and qid[di] and qnd[dn]:  
            rc[col] = fil  
            qr[fil] = qid[di] = qnd[dn] = False  
  
            if col < 7:  
                res = intend(col + 1)  
                if not res:  
                    qr[fil] = qid[di] = qnd[dn] = True  
                    rc[col] = -1  
            else:  
                res = True  
    return res
```

Seleccione una o más de una:

- ☐ a. El algoritmo de resolución del problema nunca analiza las diagonales del tablero (solo las filas y las columnas), y de allí deduce las soluciones correctas.
- ☐ b. Se usan cuatro arreglos unidimensionales para llevar el control de las casillas disponibles y un quinto arreglo para representar la solución.
- ☒ c. Para mostrar todas las posibles soluciones básicas o simétricas, básicamente solo hay que cambiar el ciclo while y  ¡Correcto! la bandera usada para cortar al encontrar una solución, por un for que obligatoriamente recorra las ocho filas en cada una de las ocho columnas.
- ☐ d. Una invocación de la forma intend(5) significa que se va a intentar posicionar en alguna fila a la reina de la columna 5.

Las respuestas correctas son:

Una invocación de la forma intend(5) significa que se va a intentar posicionar en alguna fila a la reina de la columna 5.,

Para mostrar todas las posibles soluciones básicas o simétricas, básicamente solo hay que cambiar el ciclo while y la bandera usada para cortar al encontrar una solución, por un for que obligatoriamente recorra las ocho filas en cada una de las ocho columnas.


Pregunta 17

Correcta

Puntúa 1 sobre 1

Considere el problema del Cambio de Monedas analizado en clases, y la solución mediante un Algoritmo Ávido también presentada en clases ¿Cuáles de las siguientes afirmaciones **son ciertas** en relación al problema y al algoritmo citado? (Más de una respuesta puede ser cierta, por lo que marque todas las que considere correctas...)

Seleccione una o más de una:

- ☐ a. El Algoritmo Ávido sugerido para el problema del Cambio de Monedas funciona correctamente para cualquier conjunto de valores nominales de monedas, siempre y cuando ese conjunto incluya a la moneda de 1 centavo.
- ☒ b. Sea cual sea el algoritmo que se emplee, es exigible que exista la moneda de 1 centavo, pues de otro modo no  ¡Correcto!
habrá solución posible para muchos valores de cambio.
- ☐ c. El Algoritmo Ávido sugerido para el Problema del Cambio de Monedas falla si el valor x a cambiar tiene una moneda igual a x en el conjunto de valores nominales: en ese caso, el algoritmo provoca un error de runtime y se interrumpe.
- ☐ d. Si el Problema de Cambio de Monedas no puede resolverse en forma óptima para un conjunto dado de monedas que incluya a la de 1 centavo, mediante el Algoritmo Ávido propuesto, entonces el problema no tiene solución.

¡Correcto!

La respuesta correcta es:

Sea cual sea el algoritmo que se emplee, es exigible que exista la moneda de 1 centavo, pues de otro modo no habrá solución posible para muchos valores de cambio.




Pregunta 18

Correcta

Puntúa 1 sobre 1

Considere el problema del Cambio de Monedas analizado en clases, y la solución mediante Programación Dinámica también presentada en clases ¿Cuáles de las siguientes afirmaciones **son ciertas** en relación al problema y al algoritmo citado? (Más de una respuesta puede ser cierta, por lo que marque todas las que considere correctas...)

Seleccione una o más de una:

- ☒ a. En el algoritmo basado en Programación Dinámica los valores de las monedas que sean mayores a x , son dejados  ¡Correcto!
de lado y la recurrencia de cálculo no se aplica sobre ellos.
- ☒ b. En el algoritmo basado en Programación Dinámica no es importante si el arreglo *coins* está ordenado o  ¡Correcto!
desordenado: funcionará correctamente de todas formas
- ☒ c. En el algoritmo basado en Programación Dinámica, el resultado final a retornar es el que haya quedado  ¡Correcto!
almacenado en la casilla x del arreglo *prev* que contiene los resultados intermedios (o sea, en *prev[x]*).
- ☐ d. En el algoritmo basado en Programación Dinámica, el resultado final a retornar es igual a la suma o acumulación de todos los valores almacenados en el arreglo *prev* donde se almacenaron los resultados intermedios.

¡Correcto!

Las respuestas correctas son:

En el algoritmo basado en Programación Dinámica no es importante si el arreglo *coins* está ordenado o desordenado: funcionará correctamente de todas formas,

En el algoritmo basado en Programación Dinámica los valores de las monedas que sean mayores a x , son dejados de lado y la recurrencia de cálculo no se aplica sobre ellos.,

En el algoritmo basado en Programación Dinámica, el resultado final a retornar es el que haya quedado almacenado en la casilla x del arreglo *prev* que contiene los resultados intermedios (o sea, en *prev[x]*).

Pregunta **19**

Correcta

Puntúa 1 sobre 1

Suponga que para un problema dado se ha planteado un algoritmo basado en divide y vencerás cuya estructura lógica esencial es la siguiente:

```
proceso(partición):  
    adicional() [O(n)]  
    proceso(partición/3)  
    proceso(partición/3)  
    proceso(partición/3)
```

¿Cuál de las siguientes es la expresión en *notación Big O* del tiempo de ejecución para este proceso? (Aclaración: la base del logaritmo no es esencial en una expresión en notación Big O, pero en este caso es relevante a los efectos del planteo conceptual de la pregunta).

Seleccione una:

- ☐ a. $O(n^2 * \log_3(n))$
- ☒ b. $O(n * \log_3(n))$
- ☐ c. $O(n^2 * \log_2(n))$
- ☐ d. $O(n * \log_2(n))$

✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

$O(n * \log_3(n))$

Pregunta **20**

Correcta

Puntúa 1 sobre 1

¿Cuáles de las siguientes son ciertas respecto de la estrategia de resolución de problemas conocida como *Algoritmos Ávidos*?

Seleccione una o más de una:

- ☒ a. Normalmente, una ventaja de un algoritmo ávido es que, si la regla local aplicada es correcta, suele ser sencillo de plantear y de comprender, además de veloz para ejecutar. ✓ ¡Correcto!
- ☒ b. Se trata de un planteo basado en identificar una regla local que intuitivamente parece correcta, y aplicarla una y otra vez sin volver atrás ni analizar caminos alternativos, hasta llegar a la solución del problema global. ✓ ¡Correcto!
- ☐ c. Se trata de un planteo basado en explorar y analizar cada camino posible para resolver un problema, de forma que si detecta que algún camino conduce a una solución incorrecta, se abandone ese camino y se regrese para explorar otros posibles, hasta dar con la solución correcta.
- ☐ d. Normalmente, una ventaja de intentar aplicar un algoritmo ávido es que si bien la validez de la regla local debe ser demostrada formalmente, eso es comúnmente fácil de hacer.

¡Correcto!

Las respuestas correctas son:

Se trata de un planteo basado en identificar una regla local que intuitivamente parece correcta, y aplicarla una y otra vez sin volver atrás ni analizar caminos alternativos, hasta llegar a la solución del problema global.,

Normalmente, una ventaja de un algoritmo ávido es que, si la regla local aplicada es correcta, suele ser sencillo de plantear y de comprender, además de veloz para ejecutar.

◀ [Planillas de Notas y Condición Final \(Provisorias\)](#)

Ir a...