

[Página Principal](#) / [Mis cursos](#) / [AED \(ExFi-2023/11/30\)](#) / [Contenidos](#) / [Cuestionario Teórico \(Regulares y Promocionados - 2022 o anteriores\)](#)

Comenzado el	jueves, 30 de noviembre de 2023, 08:40
Estado	Finalizado
Finalizado en	jueves, 30 de noviembre de 2023, 09:16
Tiempo empleado	35 minutos 51 segundos
Puntos	14/20
Calificación	7 de 10 (72%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

Suponga que se quiere plantear una **definición recursiva** del concepto de **bosque**. ¿Cuál de las siguientes propuestas generales es correcta y constituye la mejor definición?

Seleccione una:

- ☐ a. Un bosque es un conjunto de árboles que puede estar vacío, o puede contener n árboles (con $n > 0$).
- ☐ b. Un bosque es un bosque.
- ☒ c. Un bosque es un conjunto de árboles que puede estar vacío, o puede contener uno o más árboles agrupados con otro bosque. ✔ ¡Correcto!
- ☐ d. Un bosque es un conjunto que puede contener uno o más árboles agrupados con otro bosque.

✔ ¡Correcto!

La respuesta correcta es:

Un bosque es un conjunto de árboles que puede estar vacío, o puede contener uno o más árboles agrupados con otro bosque.


Pregunta 2

Correcta

Se 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 una invocación a si misma en su bloque de acciones, y antes de esas invocaciones  ¡Correcto! 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.
- ☐ b. La función deben tener al menos una invocación a si misma en su bloque de acciones.
- ☐ c. 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.
- ☐ d. 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.

¡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 3

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes situaciones de programación es la principal aplicación de una *Pila*?

Seleccione una:

- ☐ a. Procesar una secuencia de datos en orden de menor a mayor.
- ☒ b. Procesar una secuencia de datos en orden inverso al de su entrada.  ¡Correcto!
- ☐ c. Procesar una secuencia de datos en orden aleatorio.
- ☐ d. Procesar una secuencia de datos en el mismo orden en el que ingresaron.

¡Correcto!

La respuesta correcta es:

Procesar una secuencia de datos en orden inverso al de su entrada.

Pregunta 4

Parcialmente correcta

Se puntúa 0 sobre 1

¿Cuáles de las siguientes *son correctas* en cuanto a los tiempos de ejecución de los algoritmos de ordenamiento clásicos? (Más de una puede ser cierta... marque TODAS las que considere válidas)

Seleccione una o más de una:

- ☐ a. Algoritmos directos o simples: $O(n^2)$ en el peor caso para todos ellos.
- ☒ b. Algoritmo Quick Sort: $O(n \cdot \log(n))$ en el caso promedio, pero $O(n^2)$ en el peor caso. ✓ ¡Correcto!
- ☐ c. Algoritmo Heap Sort: $O(n \cdot \log(n))$ tanto para el caso promedio como para el peor caso.
- ☐ d. Algoritmo Shell Sort: $O(n^2)$ en el peor caso para la serie de incrementos decrecientes vista en clase.

Las respuestas correctas son:

Algoritmos directos o simples: $O(n^2)$ en el peor caso para todos ellos.,

Algoritmo Quick Sort: $O(n \cdot \log(n))$ en el caso promedio, pero $O(n^2)$ en el peor caso.,

Algoritmo Heap Sort: $O(n \cdot \log(n))$ tanto para el caso promedio como para el peor caso.

Pregunta 5

Correcta

Se puntúa 1 sobre 1

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

Seleccione una:

- ☐ a. Tienen muy mal rendimiento en tiempo de ejecución, cualquiera sea el tamaño n del arreglo.
- ☒ b. Tienen muy mal rendimiento en tiempo de ejecución si el tamaño n del arreglo es grande o muy grande, y un ✓ ¡Correcto! rendimiento aceptable si n es pequeño.
- ☐ c. Tienen muy buen rendimiento en tiempo de ejecución, cualquiera sea el tamaño n del arreglo.
- ☐ d. Tienen muy mal rendimiento en tiempo de ejecución si el tamaño n del arreglo es pequeño, y un rendimiento aceptable si n es grande o muy grande.

¡Correcto!

La respuesta correcta es:

Tienen muy mal rendimiento en tiempo de ejecución si el tamaño n del arreglo es grande o muy grande, y un rendimiento aceptable si n es pequeño.

Pregunta 6

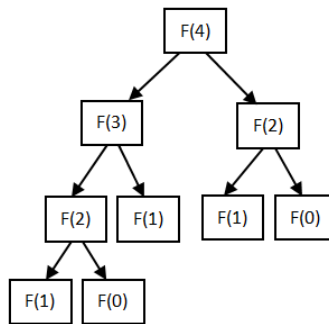
Correcta

Se puntúa 1 sobre 1

Sabemos que la recursión es una de las técnicas o estrategias básicas para el planteo de algoritmos y que una de sus ventajas es que permite el diseño de algoritmos compactos y muy claros, aunque al costo de usar algo de memoria extra en el stack segment y por consiguiente también un algo de tiempo extra por la gestión del stack. Algunos problemas pueden plantearse en forma directa procesando recursivamente diversos subproblemas menores. Tal es el caso de la *sucesión de Fibonacci*, en la cual cada término se calcula como la suma de los dos inmediatamente anteriores [esto se expresa con la siguiente relación de recurrencia: $F(n) = F(n-1) + F(n-2)$ con $F(0) = 0$ y $F(1) = 1$]. La función sencilla que mostramos a continuación que calcula el término n-ésimo de la sucesión en forma recursiva:

```
def fibo(n):
    if n <= 1:
        return 1
    return fibo(n-1) + fibo(n-2)
```

Sin embargo, un inconveniente adicional es que la *aplicación directa* de *dos o más invocaciones recursivas* en el planteo de un algoritmo podría hacer que un mismo subproblema se resuelva más de una vez, incrementando el tiempo total de ejecución. Por ejemplo, para calcular fibo(4) el árbol de llamadas recursivas es el siguiente:



y puede verse que en este caso, se calcula **2 veces** el valor de **fibo(2)**, **3 veces** el valor de **fibo(1)** y **2 veces** el valor de **fibo(0)**... con lo que la cantidad de llamadas a funciones para hacer *más de una vez el mismo trabajo* es de **7** (= 2 + 3 + 2).

Suponga que se quiere calcular el valor del sexto término de la sucesión. **Analice el árbol de llamadas recursivas que se genera al hacer la invocación $t = \text{fibo}(6)$ ¿Cuántas veces en total la función **fibo()** se invoca para hacer más de una vez el mismo trabajo, SIN incluir en ese conteo a las invocaciones para obtener **fibo(0)** y **fibo(1)**?**

Seleccione una:

- ☐ a. 0
- ☐ b. 11
- ☐ c. 12
- ☒ d. 10 ✔ ¡Correcto! Efectivamente: **fibo(4)** se invoca 2 veces, **fibo(3)** se invoca 3 veces y **fibo(2)** se invoca 5 veces...

¡Correcto!

La respuesta correcta es: 10

Pregunta 7

Parcialmente correcta

Se puntúa 1 sobre 1

Para cada uno de los algoritmos o procesos listados en la columna de la izquierda, seleccione la expresión de orden que mejor describe su tiempo de ejecución en el peor caso.

Dado un arreglo v con n componentes, buscar un valor x aplicando búsqueda secuencial.

 $O(n)$ 

Dadas dos matrices de orden $n \times n$, obtener la matriz producto aplicando el método tradicional.

 $O(n^2)$ (léase: "orden n al cuadrado")

Dado un arreglo ya ordenado v con n componentes, buscar un valor x aplicando búsqueda binaria.

 $O(\log(n))$ 

Dado un arreglo v con n componentes, acceder y cambiar el valor del casillero $v[k]$ (con $0 \leq k \leq n-1$).

 $O(1)$ 

Dado un arreglo v con n componentes, ordenarlo de menor a mayor mediante el algoritmo de selección directa.

 $O(n^3)$ (léase: "orden n al cubo")

Ha seleccionado correctamente 3.

La respuesta correcta es:

Dado un arreglo v con n componentes, buscar un valor x aplicando búsqueda secuencial. $\rightarrow O(n)$,

Dadas dos matrices de orden $n \times n$, obtener la matriz producto aplicando el método tradicional. $\rightarrow O(n^3)$ (léase: "orden n al cubo"),

Dado un arreglo ya ordenado v con n componentes, buscar un valor x aplicando búsqueda binaria.

$\rightarrow O(\log(n))$,

Dado un arreglo v con n componentes, acceder y cambiar el valor del casillero $v[k]$ (con $0 \leq k \leq n-1$). $\rightarrow O(1)$,

Dado un arreglo v con n componentes, ordenarlo de menor a mayor mediante el algoritmo de selección directa.

$\rightarrow O(n^2)$ (léase: "orden n al cuadrado")

Pregunta 8

Correcta

Se puntúa 1 sobre 1

¿Cuáles de las siguientes afirmaciones son correctas en relación a conceptos elementales del análisis de algoritmos? (Más de una respuesta puede ser válida, por lo que marque todas las que considere correctas).

Seleccione una o más de una:

- ☐ a. La notación Big O se usa para expresar el rendimiento de un algoritmo en términos de una función que imponga una cota inferior para ese algoritmo en cuanto al factor medido (tiempo o espacio de memoria).
- ☒ b. Los dos factores de eficiencia más comúnmente utilizados en el análisis de algoritmos son el tiempo de ejecución ✓ ¡Correcto!
de un algoritmo y el espacio de memoria que un algoritmo emplea.
- ☒ c. El análisis del *caso promedio* es aquel en el cual se estudia el comportamiento de un algoritmo cuando debe ✓ ¡Correcto!
procesar una configuración de datos que llegan en forma aleatoria.
- ☒ d. El análisis del *peor caso* es aquel en el cual se estudia el comportamiento de un algoritmo cuando debe procesar ✓ ¡Correcto!
la configuración más desfavorable posible de los datos que recibe.

¡Correcto!

Las respuestas correctas son:

Los dos factores de eficiencia más comúnmente utilizados en el análisis de algoritmos son el tiempo de ejecución de un algoritmo y el espacio de memoria que un algoritmo emplea.,

El análisis del *peor caso* es aquel en el cual se estudia el comportamiento de un algoritmo cuando debe procesar la configuración más desfavorable posible de los datos que recibe.,

El análisis del *caso promedio* es aquel en el cual se estudia el comportamiento de un algoritmo cuando debe procesar una configuración de datos que llegan en forma aleatoria.

Pregunta 9

Correcta

Se puntúa 1 sobre 1

¿Cuál es la diferencia entre la *abstracción de datos* y la *abstracción funcional*?

Seleccione una:

- ☐ a. Ninguna. Son sólo dos formas de referirse al mecanismo de abstracción.
- ☐ b. No existe un mecanismo de abstracción de datos ni un mecanismo de abstracción funcional. Existe sólo un mecanismo de abstracción, sin dividir en abstracción de datos y abstracción funcional.
- ☐ c. La abstracción de datos busca captar el conjunto de procesos relevante para el tipo abstracto que se quiere implementar, mientras que la funcional busca determinar el conjunto de datos más relevante para implementar ese tipo.
- ☒ d. La abstracción de datos busca captar el conjunto de datos más relevante para representar un tipo abstracto, ✓ ¡Correcto!
mientras que la funcional busca determinar el conjunto de procesos relevante para esos datos.

¡Correcto!

La respuesta correcta es:

La abstracción de datos busca captar el conjunto de datos más relevante para representar un tipo abstracto, mientras que la funcional busca determinar el conjunto de procesos relevante para esos datos.

Pregunta 10

Incorrecta

Se puntúa 0 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. $rc = [2, 0, 7, 4, 5, 1, 6, 3]$
- ☒ b. $rc = [5, 3, 6, \times, 0, 7, 1, 4, 2]$ **Incorrecto... la solución representada en este caso, es una solución válida (de hecho, es la solución número 79 según las salidas del programa backtracking02.py incluido en la Ficha 25).**
- ☐ c. $rc = [4, 7, 3, 0, 2, 5, 1, 6]$
- ☐ d. $rc = [3, 5, 7, 0, 5, 1, 2, 4]$

Las respuestas correctas son:

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

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



Pregunta 11

Correcta

Se 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. 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.
- ☒ b. Normalmente, una ventaja de un algoritmo ávido es que, si la regla local aplicada es correcta, suele ser sencillo de  **¡Correcto!** plantear y de comprender, además de veloz para ejecutar.
- ☐ c. 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.
- ☒ d. Se trata de un planteo basado en identificar una regla local que intuitivamente parece correcta, y aplicarla una y  **¡Correcto!** otra vez sin volver atrás ni analizar caminos alternativos, hasta llegar a la solución del problema global.

¡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.

Pregunta **12**

Incorrecta

Se puntúa 0 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 **caso promedio**? (Es decir: ¿Cuál es la *altura* de ese árbol en el **caso promedio**?)

Seleccione una:

- ☐ a. $\text{Altura} = n^2$
- ☒ b. $\text{Altura} = n * \log(n)$ ✖ Incorrecto... esa no es la altura del árbol, sino el tiempo total de ejecución en el caso promedio... y justamente ese tiempo depende de la altura del árbol...
- ☐ c. $\text{Altura} = n$
- ☐ d. $\text{Altura} = \log(n)$

La respuesta correcta es:

 $\text{Altura} = \log(n)$ Pregunta **13**

Correcta

Se puntúa 1 sobre 1

¿Cuál de los siguientes es el creador del famoso algoritmo de ordenamiento conocido como **Quicksort**?

Seleccione una:

- ☐ a. Donald Shell
- ☒ b. Charles Antony Richard Hoare ✔ ¡Correcto!
- ☐ c. Edsger Wybe Dijkstra
- ☐ d. J. W. J. Williams

¡Correcto!

La respuesta correcta es:

Charles Antony Richard Hoare

Pregunta **14**

Incorrecta

Se puntúa 0 sobre 1

Analice el siguiente esquema de una función en Python:

```
def procesar(n, m):  
    for i in range(n+1):  
        for j in range(m+1):  
            # ... acciones sencillas a realizar...  
            # ... suponga que no hay otro ciclo aquí...  
            # ... y que sólo aparecen operaciones de tiempo constante...
```

¿Cuál de las siguientes expresiones de orden describe mejor el tiempo de ejecución de esta función en el peor caso?

Seleccione una:

- ☒ a. $O(n \cdot n)$ ❌ Incorrecto... Esta expresión sólo sería correcta si $n == m$ pero no se supuso ese caso...
- ☐ b. $O(n \cdot m)$
- ☐ c. $O(n)$
- ☐ d. $O(m)$

La respuesta correcta es:

$O(n \cdot m)$

Pregunta **15**

Correcta

Se puntúa 1 sobre 1

¿Por qué es considerada una *mala idea* tomar como pivot al *primer elemento* (o al *último*) de cada partición al implementar el algoritmo *Quicksort*?

Seleccione una:

- ☐ a. No es cierto que sea una mala idea. Ambas alternativas son tan buenas como cualquier otra.
- ☐ b. Porque de esa forma aumenta el riesgo de caer en el peor caso, o aproximarse al peor caso, si el arreglo estuviese completamente desordenado.
- ☐ c. Porque de esa forma aumenta el riesgo de caer en el peor caso, o aproximarse al peor caso, si el tamaño n del arreglo fuese muy grande.
- ☒ d. Porque de esa forma aumenta el riesgo de caer en el peor caso, o aproximarse al peor caso, si el arreglo estuviese ya ordenado o casi ordenado. ✅ ¡Correcto!

¡Correcto!

La respuesta correcta es:

Porque de esa forma aumenta el riesgo de caer en el peor caso, o aproximarse al peor caso, si el arreglo estuviese ya ordenado o casi ordenado.


Pregunta 16

Parcialmente correcta

Se 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  ¡Correcto! en armar suconjuntos ordenados con elementos a distancia $h > 1$ en las primeras fases, y terminar con $h = 1$ en la última.
- ☐ 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. 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 $O(n^{1.5})$.
- ☐ d. En el caso promedio, el algoritmo *Shellsort* es tan eficiente como el *Heapsort* o el *Quicksort*, con tiempo de ejecución $O(n \log(n))$.

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 **17**

Incorrecta

Se puntúa 0 sobre 1

¿Cuál de las siguientes describe mejor la idea de funcionamiento en la que está basado el algoritmo conocido como ordenamiento por *Selección Simple* o *Selección Directa* para ordenar un arreglo de n componentes?

Seleccione una:

- ☐ a. Suponer que el arreglo tiene un subconjunto inicialmente ordenado que contiene sólo al primer elemento, luego realizar n pasadas, de forma que en cada una agregue el siguiente elemento al grupo que está ordenado.
- ☒ b. Realizar n pasadas, de forma que en cada una se compare a cada elemento con el siguiente, logrando que en cada pasada los mayores vaya acomodándose al final del arreglo. ✘ Incorrecto... esa es la idea del algoritmo conocido como ordenamiento por Intercambio Directo o Bubble Sort.
- ☐ c. Realizar n pasadas, de forma que en cada una se determine el menor de los elementos analizados, y llevar ese menor a la casilla pivot.
- ☐ d. Reacomodar los n elementos del arreglo en forma aleatoria, controlar si quedó ordenado, y en caso de negativo, volver a reacomodarlos en forma aleatoria, continuando así hasta que en algún momento se obtenga un arreglo ordenado...

La respuesta correcta es:

Realizar n pasadas, de forma que en cada una se determine el menor de los elementos analizados, y llevar ese menor a la casilla pivot.

Pregunta 18

Correcta

Se puntúa 1 sobre 1

¿Cuáles de las siguientes afirmaciones **son ciertas** en referencia a las *Estrategias de Resolución de Problemas* que se citan? (Más de una respuesta puede ser cierta, por lo que marque todas las que considere correctas...)

Seleccione una o más de una:

- ☒ a. La estrategia de *Fuerza Bruta* se basa en aplicar ideas intuitivas y directas, simples de codificar, pero normalmente produce algoritmos de mal rendimiento en tiempo de ejecución y/o de espacio de memoria empleado. ✓ ¡Correcto!
- ☒ b. La técnica de Programación Dinámica se basa en calcular los resultados de los subproblemas de menor orden o tamaño que pudieran aparecer, almacenar esos resultados en una tabla, y luego re-usarlos cuando vuelvan a ser requeridos en el cálculo del problema mayor. ✓ ¡Correcto!
- ☒ c. La estrategia de *Backtracking* es de base recursiva y permite implementar soluciones de prueba y error explorando las distintas soluciones y volviendo atrás si se detecta que un camino conduce a una solución incorrecta. cuando es aplicable, es más eficiente que la Fuerza Bruta, ya que permite eliminar caminos por deducción. ✓ ¡Correcto!
- ☐ d. El empleo de la *Recursividad* para resolver un problema no es recomendable en ningún caso, debido a la gran cantidad de recursos de memoria o de tiempo de ejecución que implica.

¡Correcto!

Las respuestas correctas son:

La estrategia de *Fuerza Bruta* se basa en aplicar ideas intuitivas y directas, simples de codificar, pero normalmente produce algoritmos de mal rendimiento en tiempo de ejecución y/o de espacio de memoria empleado.,

La estrategia de *Backtracking* es de base recursiva y permite implementar soluciones de prueba y error explorando las distintas soluciones y volviendo atrás si se detecta que un camino conduce a una solución incorrecta. cuando es aplicable, es más eficiente que la Fuerza Bruta, ya que permite eliminar caminos por deducción.,

La técnica de Programación Dinámica se basa en calcular los resultados de los subproblemas de menor orden o tamaño que pudieran aparecer, almacenar esos resultados en una tabla, y luego re-usarlos cuando vuelvan a ser requeridos en el cálculo del problema mayor.

Pregunta 19

Correcta

Se puntúa 1 sobre 1

¿Qué significa decir que un algoritmo dado tiene un tiempo de ejecución **$O(1)$** ?

Seleccione una:

- ☐ a. El tiempo de ejecución siempre es de un segundo, sin importar la cantidad de datos.
- ☐ b. El tiempo de ejecución es logarítmico: a medida que aumenta el número de datos, aumenta el tiempo pero en forma muy suave.
- ☐ c. El tiempo de ejecución es lineal: si aumenta el número de datos, aumenta el tiempo en la misma proporción.
- ☒ d. El tiempo de ejecución es constante, sin importar la cantidad de datos. ✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

El tiempo de ejecución es constante, sin importar la cantidad de datos.

Pregunta **20**

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes situaciones haría que el algoritmo *Quicksort* degenera en su peor caso en cuanto al tiempo de ejecución, de orden n^2 ?

Seleccione una:

- ☐ a. El algoritmo Quicksort no tiene un peor caso $O(n^2)$. Su tiempo de ejecución siempre es $O(n \log(n))$.
- ☒ b. Que el arreglo de entrada tenga sus elementos dispuestos de tal forma que cada vez que se seleccione el pivot en cada partición, resulte que ese pivot sea siempre el menor o el mayor de la partición que se está procesando. ✓ ¡Correcto!
- ☐ c. Que el arreglo esté ya ordenado, pero al revés.
- ☐ d. Que el arreglo esté ya ordenado, en la misma secuencia en que se lo quiere ordenar.

¡Correcto!

La respuesta correcta es:

Que el arreglo de entrada tenga sus elementos dispuestos de tal forma que cada vez que se seleccione el pivot en cada partición, resulte que ese pivot sea siempre el menor o el mayor de la partición que se está procesando.

[◀ Avisos](#)[Enunciado Examen Final Práctico \(Regulares - Todos los años\) ▶](#)