

Comenzado el lunes, 10 de septiembre de 2018, 21:01

Estado Finalizado

Finalizado en lunes, 10 de septiembre de 2018, 21:30

Tiempo empleado 29 minutos 53 segundos

Calificación 9 de 10 (92%)

Pregunta 1

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

¡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 2

Correcta

Puntúa 1 sobre 1

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

Seleccione una:

- ☐ a.
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...
- ☐ b.
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.
- ☒ c.
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. ✓
¡Correcto!
- ☐ d.
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.

¡Correcto!

La respuesta correcta es:

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.

Pregunta 3

Correcta

Puntúa 1 sobre 1

¿Cuál de las siguientes describe mejor la idea de funcionamiento en la que está basado el algoritmo conocido como ordenamiento por *Intercambio Directo* u *Ordenamiento de Burbuja* para ordenar *de menor a mayor* un arreglo de n componentes?

Seleccione una:

- ☒ a.
Realizar n pasadas, de forma que en cada una se compare a cada elemento con el siguiente, logrando que en cada pasada los mayores vayan acomodándose al final del arreglo. ✓
¡Correcto!
- ☐ b.
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.
- ☐ c.
Reacomodar los n elementos del arreglo en forma aleatoria, controlar si quedó ordenado, y en caso negativo, volver a reacomodarlos en forma aleatoria, continuando así hasta que en algún momento se obtenga un arreglo ordenado...
- ☐ d.
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.

¡Correcto!

La respuesta correcta es:

Realizar n pasadas, de forma que en cada una se compare a cada elemento con el siguiente, logrando que en cada pasada los mayores vayan acomodándose al final del arreglo.

Pregunta 4

Correcta

Puntúa 1 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.
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...
- ☐ 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.
- ☒ 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. ✓
¡Correcto!
- ☐ d.
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.

¡Correcto!

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 5

Correcta

Puntúa 1 sobre 1

¿Por qué motivo el algoritmo Bubblesort para ordenamiento de un arreglo usa una *bandera de corte* en la versión presentada en las fichas de clase?

Seleccione una:

- ☐ a.
La bandera de corte se usa para determinar si el ordenamiento debe hacerse de menor a mayor (bandera = True) o de mayor a menor (bandera = False)
- ☐ b.
La bandera de corte se usa para garantizar que el arreglo quede ordenado.
- ☒ c.
La bandera de corte se usa para terminar el proceso apenas se detecte que en la pasada actual no hubo intercambios, para ahorrar tiempo. ✓
¡Correcto!
- ☐ d.
No es cierto que la versión vista en clases use una bandera de corte.

¡Correcto!

La respuesta correcta es:

La bandera de corte se usa para terminar el proceso apenas se detecte que en la pasada actual no hubo intercambios, para ahorrar tiempo.

Pregunta 6

Correcta

Puntúa 1 sobre 1

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

Seleccione una:

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

¡Correcto!

La respuesta correcta es:

Charles Antony Richard Hoare

Pregunta 7

Parcialmente
correcta

Puntúa 1 sobre 1

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

Seleccione una o más de una:

- ☐ a.
El algoritmo Shell Sort consiste en una mejora del algoritmo de Selección Directa, basada 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 \log_2 usqueda del menor en cada vuelta se haga en tiempo logarítmico.
- ☐ b.
Una muy buena serie de incrementos decrecientes a usar, es la serie $h = \{...16, 8, 4, 2, 1\}$
- ☐ c.
El algoritmo Shell Sort es complejo de analizar para determinar su rendimientos en forma matemática, ya que ese rendimiento depende fuertemente de la serie de incrementos decreciente que se haya seleccionado.
- ☒ d.
El algoritmo Shell Sort 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!

Las respuestas correctas son:

El algoritmo Shell Sort 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 Shell Sort es complejo de analizar para determinar su rendimientos en forma matemática, ya que ese rendimiento depende fuertemente de la serie de incrementos decreciente que se haya seleccionado.

Pregunta 8

Correcta

Puntúa 1 sobre 1

¿Cuál es el problema si en el algoritmo *Shellsort* se elige una serie de incrementos decrecientes de la forma { ..., 16, 8, 4, 2, 1 } ?

Seleccione una:

- ☐ a.
El arreglo no quedará ordenado al final.
- ☐ b.
Ningún problema: esa serie es tan buena como cualquier otra.
- ☒ c.
Los subconjuntos analizados contendrán casi los mismos elementos cuando la distancia usada sea cada vez menor, sin garantías de lograr una buena organización del arreglo antes de la última pasada. ✓
¡Correcto!
- ☐ d.
No sólo no hay ningún problema, sino que esa serie es la mejor posible para el algoritmo Shellsort.

¡Correcto!

La respuesta correcta es:

Los subconjuntos analizados contendrán casi los mismos elementos cuando la distancia usada sea cada vez menor, sin garantías de lograr una buena organización del arreglo antes de la última pasada.

Pregunta 9

Correcta

Puntúa 1 sobre 1

Sabemos que en el *algoritmo de Shell* se termina haciendo una última pasada sobre el arreglo con incremento de compración **$h = 1$** ¿Cuál de las siguientes es cierta respecto de esa última pasada con **$h = 1$** ?

Seleccione una:

- ☐ a.
La pasada con $h = 1$ es obligatoria pero no es necesario que sea la última.
- ☒ b.
Con $h = 1$ el algoritmo se convierte en un ordenamiento por inserción simple, y sólo entonces garantiza que el arreglo quede ordenado. ✓
¡Correcto!
- ☐ c.
Con $h = 1$ el algoritmo sólo controla si el arreglo está ya ordenado, y en caso de no estarlo relanza el proceso con otra sucesión de valores h .
- ☐ d.
No es obligatorio que lo haga, pero favorece un ordenamiento más rápido.

¡Correcto!

La respuesta correcta es:

Con $h = 1$ el algoritmo se convierte en un ordenamiento por inserción simple, y sólo entonces garantiza que el arreglo quede ordenado.

Pregunta 10

Parcialmente
correcta

Puntúa 1 sobre 1

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

Seleccione una o más de una:

- ☐ a.
El algoritmo Heap Sort se basa en encontrar sucesivamente el menor (o el mayor) de entre los elementos que quedan, para llevar ese valor a su casillero final, pero de forma que la búsqueda del menor (o el mayor) en cada vuelta se haga en forma muy veloz.
- ☒ b.
El algoritmo Heap Sort arma el heap o grupo de ordenamiento con el que se ordena el vector, pero lo hace en el mismo vector, sin usar memoria extra. ✓
¡Correcto!
- ☐ c.
El algoritmo Heap Sort utiliza una cantidad de memoria adicional igual al tamaño del arreglo, para armar el heap o grupo de ordenamiento con el que se ordena el vector.
- ☒ d.
El algoritmo Heap Sort tiene es muy eficiente en tiempo de ejecución, tanto para el caso promedio como para el peor caso. ✓
¡Correcto! Por ahora, no se preocupe del significado profundo de "caso promedio" o "peor caso"...

Revise la Ficha 17, sección de Temas Avanzados...

Ha seleccionado correctamente 2.

Las respuestas correctas son:

El algoritmo Heap Sort tiene es muy eficiente en tiempo de ejecución, tanto para el caso promedio como para el peor caso.,

El algoritmo Heap Sort arma el heap o grupo de ordenamiento con el que se ordena el vector, pero lo hace en el mismo vector, sin usar memoria extra.,

El algoritmo Heap Sort se basa en encontrar sucesivamente el menor (o el mayor) de entre los elementos que quedan, para llevar ese valor a su casillero final, pero de forma que la búsqueda del menor (o el mayor) en cada vuelta se haga en forma muy veloz.