

Pregunta 1

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

Son características del algoritmo de clasificación SHELLSORT:

Seleccione una:

- ☐ a. no es estable, el orden del tiempo de ejecución es casi N por logaritmo de N en el caso promedio y cuadrático en el peor caso, y es necesaria una cantidad de memoria extra igual a la cantidad de datos de entrada
- ☐ b. es estable, el orden del tiempo de ejecución es casi N por logaritmo de N en el caso promedio y cuadrático en el peor caso, y no es necesario el uso de memoria extra
- ☒ c. no es estable, el orden del tiempo de ejecución es casi N por logaritmo de N en el caso promedio y lineal en el mejor caso, y no es necesario el uso de memoria extra
- ☐ d. es estable, el orden del tiempo de ejecución es casi N por logaritmo de N en el caso promedio y en el peor caso, y no es necesario usar memoria extra



Comprobar

Correcta

Puntos para este envío: 4,00/4,00.

Pregunta 2

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

El algoritmo de clasificación RADIXSORT tiene las características siguientes :

Seleccione una:

- ☐ a. es estable, es necesaria una cantidad de memoria extra igual a la cantidad de datos de entrada para concatenar las listas y el orden del tiempo de ejecución es casi N por logaritmo de N en todos los casos
- ☐ b. es estable, es necesaria una cantidad de memoria extra igual a la cantidad de datos de entrada para concatenar las listas, y el orden del tiempo de ejecución es casi N por logaritmo de N en el caso promedio y cuadrático en el peor caso
- ☒ c. es estable, solo es necesario memoria extra para las urnas, y el orden del tiempo de ejecución es lineal en todos los casos
- ☐ d. es estable, solo es necesario memoria extra para las urnas, y el orden del tiempo de ejecución es N por logaritmo de N en todos los casos



Comprobar

Correcta

Puntos para este envío: 4,00/4,00.

Pregunta **3**

Correcta

Puntúa 1,33 sobre 4,00

🚩 Marcar pregunta

En la siguiente implementación del método de clasificación RadixSort, "k" representa

procedure RadixSort

Comienzo

para $i := k$ hasta 1

para cada valor v de tipo t_i Vaciar($Bi[v]$) fin para

para cada registro R de A

mover R desde A hasta el final de la urna $Bi[v]$

fin para

para cada valor v de tipo t_i , de menor a mayor, hacer

concatena $Bi[v]$ en el extremo de A

fin para

fin para

Fin

Seleccione una:

- ☒ a. la cantidad de dígitos de la clave
- ✓
- ☐ b. el rango del dígito menos significativo
- ☐ c. el rango del dígito más significativo
- ☐ d. la cantidad de claves del conjunto

Comprobar

Correcta

Puntos para este envío: 4,00/4,00. Contando con los intentos anteriores, daría **1,33/4,00**.

Pregunta 4

Correcta

Puntúa 2,67 sobre 4,00

🚩 Marcar pregunta

En la siguiente implementación del método de clasificación BucketSort, "m" representa

Binsort (entrada, m)

Comienzo

urnas = nuevo array de m listas vacías

para i = 1 to n hacer

 insertar entrada[i] en urnas[DMS(array[i]).clave]

fin para

para i = 0 to m-1 hacer

 Ordenar(urnas[i])

fin para

salida = Concatenar(urnas[0]... urnas[m-1])

devolver salida

Fin

Seleccione una:

- ☒ a. el rango del dígito más significativo
- ✓
- ☐ b. la cantidad de claves del conjunto
- ☐ c. el rango del dígito menos significativo
- ☐ d. la cantidad de dígitos de la clave

Comprobar

Correcta

Puntos para este envío: 4,00/4,00. Contando con los intentos anteriores, daría **2,67/4,00**.

Pregunta 5

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

La siguiente puede ser una implementación del método de clasificación HeapSort

Seleccione una:

- ☐ a. Desde $i = N$ hasta 2 hacer
Desplaza(1, $i-1$);
Intercambia($V[1]$, $V[i]$)
Fin desde
Desde $i = N \text{ div } 2$ hasta 1 hacer
Desplaza(i , N);
Fin desde
- ☐ b. Desde $i = N \text{ div } 2$ hasta 1 hacer
Desplaza(i , N);
Fin desde
Desde $i = N$ hasta 2 hacer
Desplaza(1, $i-1$);
Intercambia($V[1]$, $V[i]$)
Fin desde
- ☒ c. Desde $i = N \text{ div } 2$ hasta 1 hacer
Desplaza(i , N);
Fin desde
Desde $i = N$ hasta 2 hacer
Intercambia($V[1]$, $V[i]$)
Desplaza(1, $i-1$);
Fin desde
- ✓ ☐ d. Desde $i = N \text{ div } 2$ hasta 1 hacer
Desplaza(1, $i-1$);
Intercambia($V[1]$, $V[i]$)
Fin desde
Desde $i = N$ hasta 2 hacer
Desplaza(i , N);
Fin desde

PREGUNTA 6

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

la siguiente secuencia de estados del vector puede haber sido producida por

71	49	140	42	122	102	95	178
49	71	42	122	102	95	140	178
49	42	71	102	95	122	140	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178

Seleccione una:

☐ a. selección directa

☒ b. burbuja



☐ c. quicksort

Pregunta 7

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

La siguiente puede ser una implementación del método de clasificación Selección Directa

Seleccione una:

- ☐ a. Desde $i = 1$ hasta $N - 1$ hacer
 $k = i$
 $\text{aux} = V[i].\text{clave}$
 Desde $j = i + 1$ hasta N hacer
 Si $V[j].\text{clave} < \text{aux}$ entonces
 $k = j$
 $\text{aux} = V[j].\text{clave}$
 Fin si
 Fin desde
 intercambia ($V[i], V[j]$)
 Fin desde
- ☐ b. Desde $i = 1$ hasta $N - 1$ hacer
 $k = i$
 $\text{aux} = V[i].\text{clave}$
 Desde $j = i + 1$ hasta N hacer
 Si $V[j].\text{clave} = \text{aux}$ entonces
 $k = j$
 $\text{aux} = V[j].\text{clave}$
 Fin si
 Fin desde
 intercambia ($V[i], V[k]$)
 Fin desde
- ☒ c. Desde $i = 1$ hasta $N - 1$ hacer
 $k = i$
 $\text{aux} = V[i].\text{clave}$
 Desde $j = i + 1$ hasta N hacer
 Si $V[j].\text{clave} < \text{aux}$ entonces
 $k = j$
 $\text{aux} = V[j].\text{clave}$
 Fin si
 Fin desde
 intercambia ($V[i], V[k]$)
 Fin desde
- ☐ d. Desde $i = 1$ hasta $N - 1$ hacer
 Si $V[j].\text{clave} < \text{aux}$ entonces
 $k = j$



Pregunta 8

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

Dado el siguiente vector de claves

71	140	49	178	42	122	102	95
----	-----	----	-----	----	-----	-----	----

la siguiente secuencia de estados del vector puede haber sido producida por

71	140	49	95	42	122	102	178
71	42	49	140	95	122	102	178
42	71	49	95	140	122	102	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178

Seleccione una:

☐ a. selección directa

☒ b. quicksort



☐ c. burbuja

Pregunta 9

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

El siguiente algoritmo de clasificación

Comienzo

for i = 1 to n do

while A[i] <> i do

intercambia(A[i], A[A[i]]);

Fin

Seleccione una:

- ☒ a. es un binsort trivial de orden lineal
- ☐ b. requiere que haya muchas claves iguales, en un rango acotado
- ☐ c. es una variante del método de intercambio y es de orden cuadrático
- ☐ d. tiene la desventaja de requerir memoria extra para ordenar

Comprobar

Correcta

Pregunta 10

Correcta

Puntúa 4,00 sobre 4,00

🚩 Marcar pregunta

Dado el siguiente vector de claves

71	140	49	178	42	122	102	95
----	-----	----	-----	----	-----	-----	----

la siguiente secuencia de estados del vector puede haber sido producida por

42	140	49	178	71	122	102	95
42	49	140	178	71	122	102	95
42	49	71	178	140	122	102	95
42	49	71	95	140	122	102	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178
42	49	71	95	102	122	140	178

Seleccione una:

- ☒ a. selección directa
- ☐ b. burbuja
- ☐ c. quicksort
- ☐ d. inserción directa

Comprobar

Correcta

Puntos para este envío: 4,00/4,00.

Pregunta **15**

Correcta

Puntúa 2,67 sobre 4,00

🚩 Marcar pregunta

Las siguientes propiedades corresponden al algoritmo de clasificación QUICKSORT:

Seleccione una:

- ☐ a. no es estable, tiene un orden del tiempo de ejecución logarítmico en el peor caso, y la profundidad de las llamadas recursivas podría llegar a ser igual a la cantidad de datos de entrada
- ☐ b. es estable, tiene un orden del tiempo de ejecución cuadrático en el peor caso, y la profundidad de las llamadas recursivas podría llegar a ser igual a la cantidad de datos de entrada
- ☒ c. no es estable, tiene un orden del tiempo de ejecución cuadrático en el peor caso, y la profundidad de las llamadas recursivas podría llegar a ser igual a la cantidad de datos de entrada
- ☐ d. es estable, tiene un orden del tiempo de ejecución logarítmico en el peor caso, y la profundidad de las llamadas recursivas podría llegar a ser igual a la cantidad de datos de entrada



Comprobar