

Comenzado el	domingo, 19 de noviembre de 2023, 18:24
Estado	Finalizado
Finalizado en	domingo, 19 de noviembre de 2023, 18:28
Tiempo empleado	3 minutos 55 segundos
Puntos	12/13
Calificación	9 de 10 (92%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

¿En qué universidad estudió, se doctoró y formó parte de su Academia de Ciencias el matemático *Wilhelm Ackerman*?

Seleccione una:

- ☐ a. Oxford
- ☒ b. Göttingen ✓
- ☐ c. Stanford
- ☐ d. Padova

Pregunta **2**

Correcta

Se puntúa 1 sobre 1

¿Cuál de los siguientes era el nombre verdadero del matemático conocido como [Fibonacci](#), que fue quien definió la famosa *Sucesión de Fibonacci*?



Seleccione una:

- ☐ a. Carl Gauss
- ☐ b. Giuseppe Peano
- ☒ c. Leonardo Pisano ✓
- ☐ d. Leonhard Euler

Pregunta **3**

Incorrecta

Se puntúa 0 sobre 1

¿Cuál es la mejora esencial que el algoritmo *Quicksort* realiza sobre el algoritmo *Bubblesort* o *Burbuja*?

Seleccione una:

- ☐ a. *Quicksort* no implementa ninguna mejora sustancial sobre *Bubblesort*.
- ☐ b. *Quicksort* acelera el cambio de posición tanto de los elementos menores como de los mayores, mientras que *Bubblesort* solo acelera a los mayores (o a los menores, dependiendo de la forma de implementación).
- ☐ c. *Quicksort* primero determina qué tan desordenado está el arreglo, mientras que *Bubblesort* procede directamente a ordenarlo
- ☒ d. En las versiones analizadas en clases, *Quicksort* sólo usa un ciclo para recorrer el arreglo, mientras que *Bubblesort* usa dos. ❌


Pregunta **4**

Correcta

Se 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)]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☐ b. `proceso(partición) :`
 `adicional() [==> O(n2)]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☐ c. `proceso(partición) :`
 `adicional() [==> O(1)]`
 `proceso(partición/2)`
 `proceso(partición/2)`
- ☐ d. `proceso(partición) :`
 `adicional() [==> O(n3)]`
 `proceso(partición/2)`
 `proceso(partición/2)`

Pregunta **5**

Correcta

Se puntúa 1 sobre 1

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

Seleccione una:

- ☒ a. 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. ✓
- ☐ b. Que el arreglo esté ya ordenado, en la misma secuencia en que se lo quiere ordenar.
- ☐ c. El algoritmo Quicksort no tiene un peor caso $O(n^2)$. Su tiempo de ejecución siempre es $O(n \log(n))$.
- ☐ d. Que el arreglo esté ya ordenado, pero al revés.

Pregunta **6**

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes estrategias de obtención del pivot es la *más recomendable* para evitar que el algoritmo *Quicksort* se degrade en su peor caso $O(n^2)$ en cuanto a su tiempo de ejecución?

Seleccione una:

- ☐ a. En cada partición, usar como pivot al valor de la última casilla.
- ☒ b. En cada partición, obtener el pivot por la mediana de tres (ya sea la mediana entre el primero, el último y el central; o bien la mediana entre tres elementos aleatorios la partición). ✓
- ☐ c. En cada partición, usar como pivot al valor de la primera casilla.
- ☐ d. En cada partición, usar como pivot al valor de la casilla central.

Pregunta **7**

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. 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.
- ☐ 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 arreglo estuviese ya ordenado o casi ordenado. ✓
- ☐ d. No es cierto que sea una mala idea. Ambas alternativas son tan buenas como cualquier otra.

Pregunta 8

Correcta

Se puntúa 2 sobre 2

Considere la función presentada en clases para calcular mediana entre el primero, el central y el último elemento de una partición del arreglo v delimitada por los elementos en las posiciones izq y der :

```
def get_pivot_m3(v, izq, der):  
    # calculo del pivot: mediana de tres...  
    central = int((izq + der) / 2)  
  
    if v[der] < v[izq]:  
        v[der], v[izq] = v[izq], v[der]  
  
    if v[central] < v[izq]:  
        v[central], v[izq] = v[izq], v[central]  
  
    if v[central] > v[der]:  
        v[central], v[der] = v[der], v[central]  
  
    return v[central]
```

El tamaño de la partición analizada es entonces $n = der - izq + 1$ elementos. ¿Cuál es el tiempo de ejecución de esta función, expresado en notación O y de acuerdo a ese valor de n ?

Seleccione una:

- ☐ a. $O(\log(n))$
- ☐ b. $O(n^2)$
- ☐ c. $O(n)$
- ☒ d. $O(1)$ ✓

Pregunta 9

Correcta

Se puntúa 2 sobre 2

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

Pregunta **10**

Correcta

Se puntúa 2 sobre 2

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

[◀ Materiales Adicionales para la Ficha 31](#)

Ir a...



[Guía 31 de Ejercicios Prácticos ▶](#)