

## RESPUESTAS DE LOSADA A ALGUNAS PREGUNTAS

hola

respuestas debajo

**11. ¿por qué se suele realizar en análisis de algoritmos centrándonos en el comportamiento para problemas de talla grande?**

- a. el análisis de algoritmos no se realiza de ese modo
- b. porque para problemas pequeños la elección del algoritmo es menos importante
- c. porque los problemas pequeños se resuelven rápidamente en todo caso
- d. porque así aseguramos el comportamiento del algoritmo en el mejor de los casos
- e. ninguna de las anteriores

la correcta es la b

es cierto que las cosas en general no son negras o blancas, y la noción de rapidez es subjetiva y dependiente de la aplicación. pero, en todo caso, hablando en términos generales los problemas de talla pequeña se resuelven rápidamente y la diferencia entre los algoritmos no es relevante para esos tiempos tan cortos.

**En otra pregunta del estilo, una de las opciones es: "porque nos permite comprender como se comportan los algoritmos en condiciones extremas", ¿esta podría ser una solución válida?**

sí, sería válida.

**12. La búsqueda con centinela respecto a la búsqueda sin centinela**

- a. mejora el orden de complejidad
- b. empeora el orden de complejidad
- c. el orden de complejidad es el mismo pero se realizan menos pasos
- d. ninguna de las anteriores

la correcta es la c. en cada iteración se hacen menos operaciones (porque has eliminado el AND y una condición). por tanto, sigue siendo un lazo lineal pero el número de operaciones en cada vuelta es más bajo.  
es decir es una recta (crecimiento lineal) pero con pendiente más aplanada.

**Asimismo, en otra pregunta similar, una de las opciones se refiere al consumo de memoria. Sabiendo que con centinela se realizan menos operaciones, ¿se podría decir también que consume menos memoria?**

NO. consume más memoria porque hay que meter una casilla más en la estructura de datos para guardar el centinela de último elemento.

**6. ¿por qué se suele realizar el análisis de algoritmos centrándonos en el comportamiento en el peor de los casos?**

- a. el análisis de algoritmos no se realiza de ese modo
- b. porque así tenemos acotado el tiempo máximo que tardará el algoritmo cualquiera que sean los datos de entrada
- c. porque el comportamiento en el peor y en el mejor de los casos suele ser equivalente
- d. ninguna de las anteriores

la correcta es la b. el peor de los casos (esto es orden superior de complejidad) te da una cota superior de la ejecución temporal del algoritmo.

**11. Dos algoritmos A y B tienen ambos orden superior lineal, el algoritmo A tiene orden inferior 1 y el algoritmo B tiene orden inferior sublineal, ¿Cuál elegimos?**

- a. A
- b. B
- c. es irrelevante pues en promedio ambos darán un rendimiento equivalente
- d. ninguna de las anteriores

la correcta es la A. son iguales en orden superior por lo que, con la información disponible, escogerías el A, que tiene mejor comportamiento en omega. otra cosa sería si te diesen más información, como orden promedio, en cuyo caso deberías decidir en base al orden promedio, que refleja más casos que el omega.

OJO: el orden promedio NO es la media entre el inferior y el superior. el orden promedio es, de la ejecución de múltiples casos, qué complejidad tendrías en promedio.

es un orden que no puedes derivar de saber el orden superior y el orden inferior. es algo que puedes derivar del análisis del algoritmo.

por ejemplo, en un caso de búsqueda lineal en promedio a lo mejor el elemento se encuentra en la mitad del array por lo que el número de pasos es  $n/2$  por lo que el orden promedio es  $O(N)$ .

- 12. El orden exacto de complejidad ( $\Theta$ ) nos da una idea de**
- a. el comportamiento del algoritmo en el mejor de los casos**
  - b. el comportamiento del algoritmo en el peor de los casos**
  - c. el comportamiento del algoritmo en el mejor de los casos y en el peor de los casos**
  - d. ninguna de las anteriores**

la correcta es la c, pues es la mas completa. cuando un algoritmo es orden exacto, por ejemplo, lineal, lo que están diciendo que es  $O(n)$  y es  $\Omega(n)$ . es decir, el orden superior e inferior son el mismo.

muchos algoritmos, como la búsqueda, no tienen el mismo orden superior que inferior y por tanto no tienen orden exacto. pero otros, donde coincide el orden superior con el inferior, tienen orden exacto (lineal, sublineal o el que sea).

- 1. Cual de las siguientes afirmaciones es cierta respecto a un Tipo de Datos primitivo convencional (p.e. float o int) frente a un Tipo Abstracto de Dato (TAD)**
- a. el TAD no tiene operaciones asociadas**
  - b. el TAD no oculta la implementación del tipo y el TD primitivo si.**
  - c. no existe ninguna diferencia entre ambos, son dos formas equivalentes de referirnos a lo mismo**
  - d. ninguna de las anteriores**

la correcta es la d. un tipo primitivo no es lo mismo que un TAD. un TAD para empezar lo puede definir el programador, mientras que los tipos primitivos ya están predefinidos.

- 3. Dadas dos implementaciones distintas de un TAD (p.e. un TAD lista implementado como un bloque contiguo en memoria o como una lista enlazada de nodos), cual de las siguientes afirmaciones es cierta:**

- a. puede ser que una implementación cumpla la especificación semántica y otra implementación no la cumple. En todo caso, ambas implementaciones serían correctas.**
- b. puede ser que una implementación cumpla la especificación sintáctica y otra implementación no la cumple. En todo caso, ambas implementaciones serían correctas.**
- c. puede ser que una implementación sea más eficiente que otra, aún realizando la misma labor**
- d. ninguna de las anteriores**

la correcta es la c