1. Respuestas Algoritmia

1.1 Análisis y diseño de algoritmos

1. ¿Qué es un algoritmo y cómo se define su eficiencia?

Respuesta: Un algoritmo es una serie de pasos que se utilizan para resolver un problema. Su eficiencia se define como la cantidad de tiempo y recursos necesarios para completar el algoritmo.

1. ¿Cuál es la complejidad temporal de un algoritmo y cómo se mide?

Respuesta: La complejidad temporal de un algoritmo es la cantidad de tiempo que tarda el algoritmo en completarse. Se mide en función del tamaño de la entrada.

1. ¿Qué es la notación big-O y cómo se utiliza para describir la complejidad temporal de un algoritmo?

Respuesta: La notación big-O es una forma de describir la cantidad de operaciones que realiza un algoritmo en función del tamaño de la entrada. Se utiliza para describir la complejidad temporal.

1. ¿Qué es la recursividad y cuándo se utiliza en el diseño de algoritmos?

Respuesta: La recursividad es un tipo de algoritmo que se llama a sí mismo repetidamente hasta que se cumpla una condición de salida. Se utiliza en algoritmos que se pueden dividir en subproblemas más pequeños.

1. ¿Cuál es la diferencia entre un algoritmo recursivo y uno iterativo?

Respuesta: Un algoritmo recursivo se llama a sí mismo repetidamente hasta que se cumpla una condición de salida, mientras que un algoritmo iterativo utiliza bucles para repetir una serie de operaciones hasta que se cumpla una condición.

1. ¿Cuál de los siguientes es un ejemplo de un algoritmo recursivo?

Respuesta: c) Torre de Hanoi

1. ¿Cuál es la complejidad temporal de la búsqueda binaria?

Respuesta: a) O(log n)

1. ¿Cuál de los siguientes es un ejemplo de un algoritmo de ordenamiento inestable?

Respuesta: d) Ordenamiento rápido

1. ¿Qué es la complejidad temporal del algoritmo de ordenamiento por inserción en el peor caso?

Respuesta: d) O(n^2)

1. ¿Cuál es el objetivo de la programación dinámica?

Respuesta: a) Encontrar la solución óptima a un problema