1. **¿ Qué significa O(n)?**
   Significa que el tiempo de ejecución crece de manera lineal con el tamaño de la entrada.

2. **¿ Cómo se calcula la complejidad temporal de un algoritmo?**
   Se calcula analizando como el tiempo de ejecución del algoritmo crece en función del tamaño de entrada.

3. **¿ Qué es búsqueda binaria?**
   Es un algoritmo para encontrar un elemento en una lista ordenada. Funciona dividiendo repetidamente la lista ordenada en dos, eliminando la mitad de los elementos en cada paso hasta que el elemento deseado es encontrado o se determina que no está en la lista. La búsqueda binaria es eficiente en listas grandes que están ordenadas.

   Complejidad Temporal: O(logn), lo que la hace significativamente más rápida que la búsqueda lineal en listas grandes.

4. **El proceso de la búsqueda binaria:**

- Ordenar los elementos del arreglo en orden ascendente o descendente.
- Definir el elemento a buscar.
- Declarar tres índices; el inicio del arreglo que sea 0 ,el indicie que esta al final y el valor medio , el cual se halla sumando los dos índices entre dos.
- Comparar el valor medio con el elemento que se quiere buscar:
  - Si es igual, se ha encontrado el elemento y se devuelve el índice.
  - Si es menor, se va descartar los elementos que están a la izquierda, para eso se debe reacondicionar los índices , el índice izquierdo será el valor medio + 1 para buscar en la mitad derecha.
  - Si es mayor, ajustar derecha a medio - 1 para buscar en la mitad izquierda.
- Finalización: Si el bucle termina sin encontrar el elemento, devolver un indicador (como -1) de que no se encontró.

5. **¿ Qué es búsqueda lineal y en que cosiste?**
   Es un algoritmo que busca elemento por elemento de una lista hasta encontrar el elemento deseado , se utiliza tanto para listas ordenadas como desordenadas, pero es más ideal para listas desordenadas o listas pequeñas.

   Complejidad Temporal: O(n), donde n es el número de elementos en la lista.