

1. e) Quick sort
Debido a que utiliza un número para ubicar a partir de este los números mayores
2. d) No está en la lista
Debido a que el mejor caso posible es que el elemento buscado este en la mitad, por otro lado, si el elemento está en la primera o la última posición, se toma de igual manera, ya que es necesario hacer el mismo proceso para ambos, por otro lado si el elemento no está en la lista, tiene que hacerse todo el proceso para llegar al caso de salida
3. a) $O(n)$
Debido a que en la búsqueda lineal o búsqueda secuencial se crea un solo for en el que se recorre toda la lista hasta que se encuentra el elemento, es por esto mismo que no es un algoritmo eficiente.
4. c) $O(n^2)$
Según lo que podemos ver en la tabla de complejidad.
5. b) $O(n \log(n))$
Según la tabla
6. d) El algoritmo de inserción intercambia elementos adyacentes, mientras que el algoritmo de selección inserta cada elemento en la posición correcta.
Debido a que lo que hace el algoritmo de inserción es comparar hacia atrás e intercambiar.
7. a) El elemento buscado está en la mitad de la lista.
Debido a que si el elemento se encuentra en la mitad, cumplirá la primera condición y directamente retorna la posición de este elemento.
8. b) $O(\log n)$
Debido a que como se partirá la lista, nunca se recorre todo el ciclo.
9. a y b)
Debido a que en cada partición de la lista se creará una lista vacía.
10. c) $O(n^2)$
Según la tabla