best searching algorithm

1. **Linear Search with complexity O(n)**
2. **Binary Search with complexity O(log n)**
3. **Search using HASH value with complexity O(1)**

**Linear Search with complexity O(n):**

1. we have to traverse the entire data structure till we find the element, so in the worst case we have to traverse till end of the DS .
2. hence the order/complexity of linear search is O(n).
3. This is a brute force way of doing it.
4. **pros:**  
   suitable for smaller sized data structures.  
   suitable for data structures which are not sorted.  
   simpler approach, simple and less code (KISS principle).
5. **cons:**  
   for large sized data structures this wont do any good in terms of time complexity

**Binary Search with complexity O(log n):**  
Binary search implements divide and conquer.

1. DS elements needed to be in sorted manner.
2. **logarithm** is just a mathematical scale to represent number system in powers of base, in binary search case, base=2

**pros:**  
Best suitable for large sized arrays

**cons:**  
array elements must be sorted prior to search

**Search using HASH value with complexity O(1):**  
Insert the elements of the data structure into a hash implemented data structure like Hashtable or HashMap and you are good to go with one line statement:  
hashArr.contains(a[i])  
Since the elements of hashmap are indexed by hashcode, the time to search for any particular element would almost be = 1 (CONSTANT time)

**pros:**  
Best in case of medium-large sized arrays

**cons:**  
If the array is very large then it might lead to collisions in the hash implemented DS  
Additional space requirements to store array elements into hashmap

Map.containsKey() considering that you are using a HashMap since searching in HashMap is done in O(1).

List.contains() generally should resort to sequential search or a binary search thus the complexity will be atleast O( n)

**WHICH ONE TO USE:**  
It is actually a trade off between these three approaches on which one to use. There is never always one best approach to follow blindly. We should analyze the scenario and adopt one of these.

Time complexity of Binary search:





