Explain how arrays are represented in memory and their advantages.

Arrays are the storage of elements into the memory locations which are in sequence, that is, it implies that all the elements are stored next to each other in the memory. You can access the individual item in the array by its index. This index is actually the position of the element within the array. This index-based access has constant time complexity, and the exact process to be the special register to the Direct Memory Addressing arithmetic.

Advantages of Arrays:

* Efficient Index-based Access: Arrays allow for immediate access to any element using its index, resulting in very fast retrieval times. The so-called direct access is one of the means achieved through a simple computation utilizing the index and the base address of the array.
* Cache Friendliness: Since array elements are distributed over successive memory locations, then they will take advantage of cache locality. The central processing unit(CPU) will load an outgoing through a high-speed master bus to the prefetcher so that when adjacent elements are requested the memory cache is ready to be read thus the job will run faster.

Analyze the time complexity of each operation (add, search, traverse, delete).

Time Complexity:

* Add: O(1) on average, but can be O(n) if the array needs to be resized (not applicable here as the array is fixed-size).
* Search: O(n) as you may need to check each element in the array.
* Traverse: O(n) as you need to visit each element.
* Delete: O(n) because you need to find the element (O(n)) and then shift subsequent elements to fill the gap (O(n)).

Discuss the limitations of arrays and when to use them.

Limitations of Arrays:

Fixed Size: Once an array is created, its size is immutable, the reason for its inability to dynamically change its size. Accordingly, it is worthless in management techniques for a variable list of elements.

Inefficient Deletions: Deleting requires to move the remaining elements, which may cause a significant performance degradation in the case of huge arrays.

Lack of Built-in Methods: Arrays, unlike higher-level data structures such as ArrayLists or Linked Lists that offer inbuilt methods for insertion, deletion or searching, do not have any extra methods.

Arrays are suitable when the size of the dataset is known and fixed, which leads to both quick calculations and a more stable computer memory. Instead, if you have a set of dynamic data such as data based on the skills of an employee, which need to be customized from time to time, then the solution is not the arrays but the ArrayLists or Linked Lists which have a built-in set of methods for insertion, deletion, or searching operations to be used.