**Limitations of Arrays**

The limitations of the array in the data structure are:

* the dimension of an array is determined the moment the array is created, and cannot be changed later on;
* the array occupies an amount of memory that is proportional to its size, independently of the number of elements that are actually of interest;
* if we want to keep the elements of the collection ordered, and insert a new value in its correct position, or remove it, then, for each such operation we may need to move many elements (on the average, half of the elements of the array); this is very inefficient.
* if the number of elements of an array is less than the allocated size of array, the memory space is wasted.

**How linked list overcome this issue?**

An array exists in memory as a contiguous block, and the location of that block is selected based on how big it needs to be. So if the array needs to grow, there's no guarantee that the adjacent memory is safe to expand into. You could be overwriting something important.

Arrays also have the property of random access indices, which means you can access any element in constant time. This is possible because the machine can take an index like 5, multiply it by the size of each segment of the array, add that to the memory address of the first element, and the result is the exact memory address of the element at index 5. This only works because the segments are contiguous. That makes it impossible to rewire the array. You can't eliminate the 5th element and have the 4th and 6th become adjacent. The closest you can get is setting the 5th element to an empty value.

A linked list abandons the random access benefit in exchange for the flexibility to resize and rewire the structure. It does this by storing each element in an arbitrary memory location. What it actually stores is a structure called a "node," which holds both the current element value and the location of the next node.

So to locate the 5th element in a linked list, you have to start at the first node and advance one node at a time. You can easily rewire the list just by changing where the nodes point. This cannot be done with an array because the order is defined by the actual location in memory, not as a secondary stored value.

In the end, the linked list overcomes the array's flexibility limitation by paying a higher memory cost and sacrificing constant-time random access.