Linked lists are preferable over arrays when:

1. you need constant-time insertions/deletions from the list (such as in real-time computing where time predictability is absolutely critical)
2. you don't know how many items will be in the list. With arrays, you may need to re-declare and copy memory if the array grows too big
3. you don't need random access to any elements
4. you want to be able to insert items in the middle of the list (such as a priority queue)

Arrays are preferable when:

1. you need indexed/random access to elements
2. you know the number of elements in the array ahead of time so that you can allocate the correct amount of memory for the array.
3. memory is a concern. Filled arrays take up less memory than linked lists. Each element in the array is just the data. Each linked list node requires the data as well as one (or more) pointers to the other elements in the linked list.