

1. linked list will require more memory. In each node of linked list, we need to store 2 variables, the value and the address of next node. While a single-dimension array only need to store one value in each cell since the address of a certain cell can be calculated from the address of the first index.
2. The two structures should take around the same time to access the first integer, since we are both taking a pointer and getting the value of the variable where the pointer is pointing at.
3. Linked list would take more time to access the 9th integer. For a single-dimension array, we can calculate the address of the 9th cell and access it directly, where a linked list need to go through the first 8 nodes to find the address of the 9th node.
4. Single-dimension array would take more time to insert an integer at the beginning, since it need to copy every single value in the array from the nth cell to (n+1)th cell, then insert the new integer. While a linked list only need to construct a new node and adjust 2 pointers (the head pointer and the next pointer of the new node).
5. Linked list would take more time to insert an integer at the end of sequence. It need to go through all nodes in the list to find the last node, and then adjust the next pointer to a new node. While a single-dimension array can find the address of the last cell quickly and put a new value there.