HW3 Guiquan Sun
3937104003

Question 1:

Passing parameters by value:
pros: a local copy has been created, which means the local variable can be operated in the function, but the source data will not be affected.
cons: a local copy has been created and will uses more memory and consume more time.

Passing using Pointers:
pros: we can manipulate the source data in another scope.
cons: we need to see whether all the value are equal to null, which are pointed by pointers, because it may be make crash occur. Another cons is if we want to access value, do dereference is necessary.

Passing using Reference:
pros: we can manipulate the source data in another scope. Another one is maybe it can only take small memory.
cons: the source data could be modified.

Passing using const Reference:
pros: the source data can not be changed. It is fast.
cons: we can not know whether the parameters change from the function.

When only passing the basic data type and the function do not change the source data, passing parameters by value.

When passing a pointer is valid and logical, passing using pointers.

When we need to modify source data, Passing using Reference is preferred.

When we do not want to modify source data, Passing using const Reference is preferred.

Question 2.

The time complexity is O(nlog(n)).

Question 3.

bool SinglyLinkedList::empty(),　the time complexity is O(1).

int SinglyLinkedList::size(),　the time complexity is O(n).

void SinglyLinkedList::push_back(int i),　the time complexity is O(n).

void SinglyLinkedList::push_front(int i),　the time complexity is O(1).

void SinglyLinkedList::insert_after(ListNode *p, int i),　the time complexity is O(1).

void SinglyLinkedList::erase(ListNode *p),　the time complexity is O(n).

void SinglyLinkedList::pop_front(),　the time complexity is O(1).

void SinglyLinkedList::pop_back(),　the time complexity is O(n).

int SinglyLinkedList::back(),　the time complexity is O(n).

int SinglyLinkedList::front(),　the time complexity is O(1).

ListNode *SinglyLinkedList::GetBackPointer(),　the time complexity is O(n).

ListNode *SinglyLinkedList::GetIthPointer(int i),　the time complexity is O(n).

void SinglyLinkedList::print(),　the time complexity is O(n).

Question 4.

The time complexity is O(n).

Question 5

The time complexity of all are O(1).

Question 6

The time complexity is O(n).