HW3 EE599

Name: Tianyi Xu

USC ID: 3860934574

Question 1

a. by value

pros: would not affect the original variable, safe(would affect the memory space)

cons: inefficient(need copy the original variable)

when: do not want to change the original variables and the original variables are simple

b. using pointers

pros: efficient(need not copy the original variable)

cons: unsafe(pointers may point to other memory space)

when: need change the variable the pointer points to in the function

c. using references

pros: efficient(need not copy the original variable), safe(would affect the memory space)

cons: may change the original variable when you do not want

when: do not mind changing the original variable

d. using const references

pros: efficient(need not copy the original variable), would not affect the original variable, safe(would affect the memory space)

cons: may bring some inconvenience to users

when: want the parameters are read-only and want the programs effective

Question 2

Iterate the vector and use hashmap to find the two numbers.

Runtime Complexity: O(N)

Question 3

a. SinglyLinkedList()

Runtime Complexity: O(1)

b. SinglyLinkedList(const std::vector<int> &inputs, int i)

Itrate the inputs and connect each item.

Runtime Complexity: O(N)

c. ~SinglyLinkedList()

Firstly use a map to check whether it is cyclic, if it is, cut the connection of cycle. Then iterate the linkedlist and delete each node.

Runtime Complexity: O(N)

d. empty()

Runtime Complexity: O(1)

e. size()

Iterate the linkedlist.

Runtime Complexity: O(N)

f. push_back(int i)

Iterate the linkedlist and find the last node. Then push i.

Runtime Complexity: O(N)

g. push_front(int i)

Runtime Complexity: O(1)

h. insert_after(ListNode* p, int i)

Runtime Complexity: O(1)

i. erase(ListNode* p)

Iterate the linkedlist and find and delete the node.

Runtime Complexity: O(N)

j. pop_front()

Runtime Complexity: O(1)

k. pop_back()

Iterate the linkedlist and find the last node. Then pop it.

Runtime Complexity: O(N)

l. back()

Iterate the linkedlist and find the last node.

Runtime Complexity: O(N)

m. front()

Runtime Complexity: O(1)

n. *GetBackPointer()

Iterate the linkedlist and find the last node.

Runtime Complexity: O(N)

o. *GetIthPointer(int i)

Iterate the linkedlist and find the ith node.

Runtime Complexity: O(N)

p. print()

Iterate the linkedlist and print each node.

Runtime Complexity: O(N)

Question 4

Iterate the input and use a stack to help.

Runtime Complexity: O(N)

Question 5

Runtime Complexity: O(1)

Question 6

a. VectorHandle()

Runtime Complexity: O(N)

b. print()

Iterate the vector and print each item.

Runtime Complexity: O(N)

c. handle(int n)

Runtime Complexity: O(N)