North South University
Electrical and Computer Engineering
CSE 225.1 Data Structure and Algorithm
Term Assignment – Spring 2022

1. Operations on Linked Lists [30 point]

Part A: Operations on Head-Only Lists [20 points]

Implement a data type List that realizes linked lists consisting of nodes with integer values. A class of type List has one field, a pointer to a Node head and with the following functions. The structure of type Node has two fields, an integer value and (a pointer to) a Node next.

The type List must have the following methods:

boolean IsEmpty()	returns true when List is empty;
int LengthIs()	returns the number of nodes in the list, which is 0 for the empty list;
void Print()	print the content of all nodes;
void AddAsHead(int i)	creates a new node with the integer and adds it to the beginning of the list;
void AddAsTail(int i)	creates a new node with the integer and adds it to the end of the list;
Node Find(int i)	returns the first node with value i;
void Reverse()	reverses the list;
int PopHead()	returns the value of the head of the list and removes the node, if the list is nonempty, otherwise returns NULL;
void RemoveFirst(int i)	removes the first node with value i;
void RemoveAll(int i)	removes all nodes with value i;
void AddAll(List l)	appends the list l to the last element of the current list, if the current list is nonempty, or let the head of the current list point to the first element of l if the current list is empty.

Part B: Operations on Head-Tail Lists [10 points]

Suppose we include another pointer in the class, and it points to the tail of the list. To accommodate and take advantage of the new pointer, we need to modify some methods. Write the new versions of the methods named as V2 where you need to manage the tail pointer. For example, if you need to modify funcOne() then add a new function funcOneV2().

CodeBlocks Details

Project Name: AsnListDemo, Files: List.h, List.cpp, main.cpp

North South University
Electrical and Computer Engineering
CSE 225.1 Data Structure and Algorithm
Term Assignment – Spring 2022

2. Operations on Hash Tables [20]

In this problem, you will implement a data structure for hashing with linear probing. Assume that we have functions for insertions, deletions, and finds. Define four functions as per description. Assume the keys are non-negative integers, which are hashed into a table of size maxSize. An empty table entry can be indicated using -2. Use -1 in a location to indicate that the entry formerly in the location has been deleted. Use hash function key%20.

a. Find(x) which returns the index of where x is stored if it is in the hash table and -1 otherwise.

b. Insert(x) which inserts x into the hash table with the precondition that x is not currently in the table. Naturally, a location with a -1 in it can be used to place a newly inserted item. Your algorithm should indicate when the hash table is full and x cannot be inserted.

c. Delete(x) which removes x from the hash table using lazy deletion.

d. Print() which prints all the valid items in the hash table.

e. Rehash() which returns a new table of the same size maxSize which contains all the items of the old table, but has no items marked as deleted. [Hint. Create a new object of HashTable, insert all the valid items from the current HashTable to the newly created one].

CodeBlocks Details

Project Name: AsnHashDemo

Files: HashTable.h HashTable.cpp main.cpp