CSCE 221 Cover Page
Programming Assignment #5
Due November 13 by midnight to CSNet

First Name Zhiyang Last Name Zeng UIN 720005338

User Name zhiyangzeng E-mail address zhiyangzeng@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero. According to the University Regulations, Section 42, scholastic dishonesty are including: acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion read more: Aggie Honor System Office

Type of sources			
People	peer teachers		
Web pages (provide URL)	stackoverflow.com, cplusplus.com		
Printed material	slides on disjoint set		
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name  Date 11/13/15

DisjointSet Report

I. Program Description and Purpose

The purpose of this assignment is to implement disjoint sets and union find using linked lists. It contains 3 files. The first file is TemplateDoublyLinkedList.h although it is a declaration and implementation file for class DListNode. The disjointset.h file is for making the sets, appending them, and finding sets. The main.cpp file is for testing the disjointset implementation.

II. Data Structure Description

The DListnode class, in addition to original data and functions found in nodes in DoublyLinkedLists, contains node pointers representative and trailer, int key and listSize, and their respective return and set methods. The disjointset class is to provide a way to union these nodes. The DisjointSet constructor declares a vector nodeLocator with size n. To avoid confusion in the output, there is also a Boolean vector to determine whether a cell in the vector should be displayed or not, depending on the union rank.

- MakeSet method allocates memory of a new node with obj T and int key, and declares a pointer to the node. The node will point representative and trailer to itself because it is a set of one. It will set the size to be one, and store its location in the nodeLocator with index key-1. The method will then return the pointer.
- FindSet method can either take a key or a node pointer as argument. If it is a node pointer, it will return the representative of the set the pointer belongs to. If it is a key, the method will get the pointer in the nodeLocator vector with index key-1, and return its representative.
- Union method unites two sets by appending the shorter list onto the longer one. The method first compares the size of representatives from each set, then designates the set with bigger or equal size the representative set. The trailer of the head set will doubly link to the representative of the tail set. Then the method will iterate through each node in the tail set to make their representative pointers point to the new representative. After all is done, the cell in the nodeLocator at which the shorter list is located will be set to invisible so that the output method will not print the cell.
- The overridden << operator print out the all the sets in a disjoint set method.

III. Runtime analysis

FindSet with node as argument will always take 2 operations and is $O(1)$. FindSet with key as argument with take 3 operations and is $O(1)$. FindSet method is always $O(1)$ in best and worst case.

Union

- In the worst case running time function is $5n_2 + 19$ where n_2 is the size of the shorter list. If the sizes of two lists are equal, Union will be $O(n)$.
- In the best case in which the shorter list is a set of 1 element, running time function is 5+19=24. It is $O(1)$.

IV. Instruction to Compile

Make all to compile. ./main to run.

V. Logical exceptions

The disjointset class checks exceptions and throws error when

1. MakeSet takes a key with index out of bounds on the nodeLocator vector
2. Trying to Union nodes in the same set
3. FindSet taking an invalid nodeKey that will result in nodeLocator index out of bounds
4. FindSet takes an invalid DListNode such as null pointer
5. DisjointSet constructor takes a negative number

VI. C++ Features

Object oriented, generic programming feature

This assignment separates into 3 files as an object oriented programming practice: a basic node that stores and retrieves information, a disjointset class that union the nodes together into a data structure, and a main testing file. It also uses template as generic programming feature. Standard exception library and operator override is used.

VII. Testing results

1. Testing results shown in the main.cpp file

After disjointset *a* initialization and make sets, there are 5 individual sets displayed with {} around separate sets (on different lines):

{a} {b} {c} {d} {e}

After calling Union on (c, d), 4 sets will be displayed:

{a} {b} {cd} {e}

After Union (d, e), 3 sets will be displayed:

{a} {b} {cde}

Findset(a) will return a. Findset(d) will return c because c is the representative of the set d belongs to.

After Union (a,b), 2 sets will be displayed:

{ab} {cde}

After Union (a,e), 1 set will be displayed:

{cdeab}

Findset(a) and Findset(e) will both return c. Listsize(a) and Listsize(e) will both return 5.

2. Other testing results not shown but are replicable

Union (d,c) after initialization will display:

{a} {b} {dc} {e}

because the ordering on 2 sets of the same size is used to determine which set becomes the representative.

Union (a, c) on {ab} {dc} will display {abdc}. Union (c, a) on {ab} {dc} will display {dcab}.
All the exception with invalid inputs have also been tested and displayed the intended error message.