

Programming Assignment #3

CSCE221 - Data Structure and Algorithms

Instructor: Mohsen Aznaveh

TEXAS A&M UNIVERSITY

Due: Wednesday, April 8, 2020 11:59 pm

PA3 Instructions

- Basic task before starting assignment
 - Download PA3.zip from piazza and Unzip it.
 - Move PA3 folder to your CSCE-UIN local GitHub repository.
 - Go to terminal, see the new changes in your git repository by running *git status* command.
 - Add new changes i.e. a new PA3 folder, using command *git add filename*
 - Make new commit of this change using *git commit -m "commit message"*
 - Push the change using *git push* command.
- Once above tasks are done, you have basically pushed the PA3 code base without solution on your GitHub.
- Now start working on the assignment, complete it and push the completed assignment on GitHub along with **makefile** and **PDF** file which contains answers to few non-coding based questions and your explanations as part of this assignment.
- The PDF can be hand-written then scanned, or LaTeX, MS Word, what-ever you prefer. But only include the one and only one final PDF, not *.docx, *.tex, etc.

Important Notes

- Following actions will led to 0 points for this assignment:
 - Manually uploading your assignment using upload option on GitHub. This can be detected.
 - No makefile submission.
 - Using g++ *.cpp in your makefile.
 - Code plagiarism and cheating. It will be checked using MOSS.

Question 1

(4.50 from the book, 20 points)

Write a program that reads a C++ source code file and outputs a list of all identifiers (that is, variable names, but not keywords, that are not found in comments or string constants) in alphabetical order. Each identifier should be output with a list of line numbers on which it occurs. (There is no need for the list of lines and number of occurrences)

In the file `q1.cpp` a part of the program is written for you. You need to write inside the loop so the set `idents` include identifiers.

For example when the input is `sample.cpp`:

```
#include <iostream>
int main()
{
    int x; int y;
    std::cout << "Howdy Ags!"<<std::endl;
    return 0;
}
```

The output will be:

```
$ ./a.out
enter the file name: sample.cpp
```

```
cout
x
y
```

Question 2

(10 points)

In the file `q2.cpp` the program reads an input file `coronamap` and make a map of countries and number of confirmed cases.

Write a loop that searches the values of the map and print out the country with maximum number of confirmed cases.

For example if we have the file `coronamap` as:

```
China 81394
France 37575
Iran 35408
Spain 72248
USA 119682
Germany 57695
Italy 92472
```

The output of the program would be something like this

```
$ ./a.out
enter the file name: coronamap
Most confirmed cases are in USA
```

Question 3

(5.3 from the book, 20 points)

Write a program to compute the number of collisions required in a long random sequence of insertions using linear probing, quadratic probing, and double hashing

The skeleton of the code is provided for you. You have to write the functions `insertLinear`, `insertQuad` and `insertQuad`.

The output would be something like

```
$ ./a.out
enter the size of the table 10000
Num linear collides 244721
Num quadratic collides 46783
Num double hash collides 2573
```

Question 4

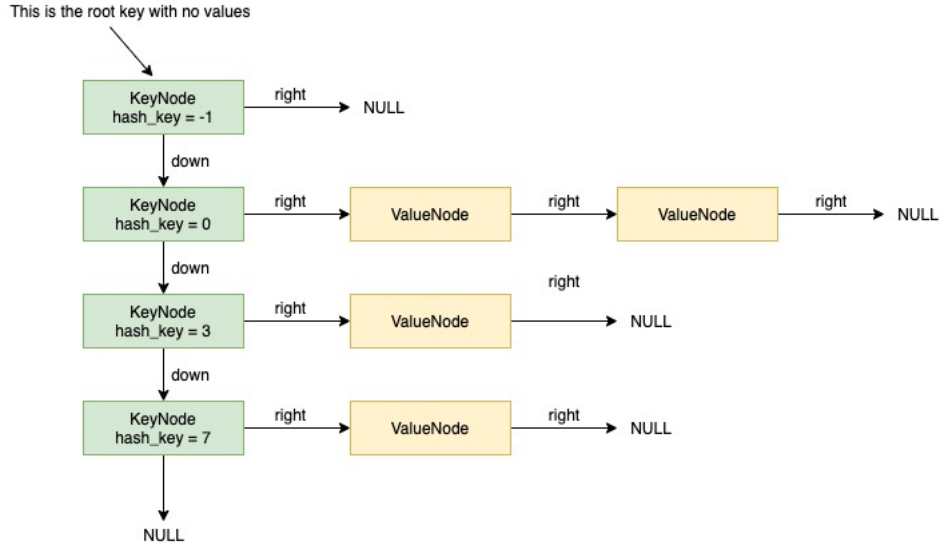
(4.1 = 20 points, 4.2 = 30 points)

`OrderedMap.h` contained the implementation of ordered map using hashing and chaining in c++ and `q4.cpp` is calling the function to test it. `KeyNode` is the key of the map which contains the hashed value of key, whereas `ValueNode` is the value node which contains both key and value in it's real form.

```
struct ValueNode
{
    Key key;
    Value value;
    ValueNode* right;
};

struct KeyNode
{
    int hash_key;
    ValueNode* right;
    KeyNode* down;
};
```

Following is a representation of this map:



Green colored boxes are the KeyNode structures and Yellow Colored boxes are ValueNode structures. By default, first key node is treated as root node with hash_key value equals to -1 and no value nodes. All the key nodes are sorted downwards with respect to the hash key value (because it is an ordered map wrt key). Every key node is linked to a linked list of value nodes which stores key/value pair and pointer to next node.

- **Lab Exercise:** Implement `int hashFunction(const Key &key)` function in `OrderedMap.h` file that will take the key object as input and return the hash value of the key in integer format. There is no restriction on the implementation of hash function. You can come up with any hash function of your own and we appreciate you exploring some ideas to implement it. **Note:** The value returned by your hash function should be between 0 and `MAP_MAX_SIZE - 1` and for this, you can use modulus operator.
- Complete the function
`void insert(int hash_key, const Key &key, const Value &value, KeyNode* t)`
 that will insert a new key value pair in the map given the map root node, key hashed value, key and value pair. **Note:** Since it is an ordered map, if a new hash_key is given, insert it in the sorted format. If there is an already existing key (chaining), insert the new value node to the end.

Below is the sample output once your code is complete. **Note:** Hash value may differ based on your hash function implementation.

```

-1      -> Root KeyNode of Map
8       -> ( Ocean , Pacific )
10      -> ( Insect , Cockroach )
11      -> ( State , Texas )
12      -> ( Vegetable , Carrot )
14      -> ( Fruit , Apple ) ( Fruit , Orange )
18      -> ( Place , Los Angeles ) ( Place , New Orleans )
  
```