CSCE 221 Cover Page
Programming Assignment #6
Due **April 29** by midnight to eCampus

First Name Arash Last Name Abdollahzadeh UIN 126005460

User Name aa75442@tamu.edu E-mail address aa75442@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	Ryan (PT) my section's TA		
Web pages (provide URL)	reading from file		
Printed material	lecture slides PA2		
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 Arash Abdollahzadeh Date 4/29/2018

Programming Assignment #6

Due ***April 29*** to submit to eCampus

- **Report**

1. Explain how your brackets operator works.

   My brackets operator works as follows: first, there is a variable called temp is created that points to an E object (int in this case). then, I check to see if the object with the key that was inserted is null or not. If it was a null, I insert the object with a 0 key, and if it wasn't null, I return the dereferenced pointer to that object.

   (a) What is the running time expressed in terms of big-Oh asymptotic notation of your brackets operator?
   The running time of my brackets operator is O(logn) if the inputs to the map(keys) are inputed in a certain order. But, since there is no way to confirm that the tree constructed by a text file would be a dense and balanced BST, the complexity is O(n) since the tree can end up looking like a linked list, but this is very unlikely.

   (b) How could you improve the running time? Here don't think about micro optimizations, think about changing data structures or major algorithm changes.
   By using a hash table with a well-performing hash function, we can obtain O(1). Now, I am just using a BST; and in the best case, I will get O(logn), but with a hash tablel, I can get O(1) on any case, if the hash table is done correctly, but it needs more memory to implement.

2. Explain what the advantages and disadvantages of implementing map with the following data structures
   (a) A vector of key_values b.
   The advantages would be that there is less overhead with using pointers and recurrsion for implementation and it is easier to debug. Disadvantages are slower search and insert (bracket) functions, they are all going to be O(n), memory consumption may increase due to doubling the size of vector (for large inputs).
   (b) A tree of key_values.
      i. Regular binary
      The advantage is the possible quicker insert and search function which is done through the bracket operator; also, since we are using pointers, there is no extra space required to be allocated, so it is more memory efficient. The disadvantage is that the iterator class is harder to implement since we have to deal with recursion now, and unless we use a balancing technique, it is not guaranteed to be better than the vector implementation.

      ii. Red Black
      The advantage will be that the O(logn) is definitly achievable for the insert and search operations (bracket operator) for any input, since we are now using tree balancing technique and the tree will be balanced after insertion.
      The disadvantages include more complication when implementing.
      iii. AVL
      The advantages and disadvantages are the same as a Red-black tree since we are just using a different technique for balancing a tree. Also, one of the additional advantage is that balancing a tree is easier and more efficient, but agian, it is more complicated to do vs. vector.

iv. 2-4 The advantages and disadvantages are the same as a Red-black tree since we are just using a different technique for balancing a tree. Additionally, one disadvantage is that it is more difficult to implement, but it reduces the runtime complexity.

(c) A Hash Table of key_values
The advantage of using a hash table is the that it significantly improves the efficiency of searching and inserting if the correct hash table implementation is used. The O(1) for searching and inserting is guaranteed (if the hashing is done correctly). The disadvantage is that it uses more memory and is harder to implement.

3. Explain one real life use case for a map, you should say specifically why it would benefit from using a map over a simpler data structure like a vector. This doesn't have to be something that exists currently, it's just an idea for how you could use a map. Example (don't use this one) storing word frequencies for a large document.

   One example for this data structure is a system to retreive customers' email address by their name. This data structure contains the key_value pair that are both strings and we can easily search for an email address by map[name] which returns the email address in O(1) (if hash table was used). Or another example is a system that checks to see if the number of times a password was entered is above a certain number, where the key is the password and the item is the number of times it was entered withing the past 24 hours.

4. Grading
   When we grade this, we will not be using your main file, we will be using our own. Because of this do not change the file name that holds the main function, also make sure you use a make file. When we grade we will be making sure that you map does not crash/throw exceptions under any circumstances. We will be taking some input from file and storing them into your map, performing some operations using the brackets operator, then printing the map out, your map should print in sorted order. We will provide a `main` function that will test all of the same operators, but with different inputs.

Points distribution	
Using PA4 Tree	15
Input/output operators	10
Constructors	5
Brackets adding	5
Brackets searching	5
Brackets search miss	5
Brackets update	5
Iterator `begin()/end()`	10
Iterator `operator++`	10
Report	30
Total	100