**UNIVERSITY OF BUEA**

**FACULTY OF SCIENCE**

**DEPARTMENT OF COMPUTER SCIENCE**

**17TH MAY 2016**

**CSC 316 IN-COURSE GROUP PROJECT**

**COURSE NAME: FUNCTIONAL PROGRAMMING**

**PROJECT NAME: COMPACT PERSISTENT INTERVAL TEMPORAL TREE**

**PROJECT NUMBER: 1.2(A AND B)**

**PARTICIPATING STUDENTS: 2**

|  |  |  |
| --- | --- | --- |
| **NAME** | **UB NUMBER** | **DEPARTMENT** |
| **AMATE SUBI YOLANDE** | **SC13A053** | **COMPUTER SCIENCE** |
| **DJIMELI KONRAD NIBA** | **SC14A305** | **COMPUTER SCIENCE** |

**COURSE INSTRUCTOR: Dr William S. Shu**

**PART 2: PROJECT DEVELOPMENT REPORT.**

**(a) Analysis process:**

- Developing an interval tree that persists over time, in which items are stored at the nodes that correspond to the inner time interval over which they persist. Intervals at levels of the tree must be non-overlapping. Also, the root nodes of all tree versions are contained in a doubly linked list in their respective time-order.

- In order to persist the tree, we would have to create a copy of every node along the path where a change is to occur. Alternatively we would have had to create a new copy of the entire tree for every new version of the tree, but this approach would have been more inefficient.

- We found no lacunae in the program requirements and so the clarifications we got from the course instructor compelled us to adopt the options we considered for analysis. We explored the notion of linked lists and referencing in the ocaml programming language , thus enabling us access the data in their respective time-order.

**(b) Design process:**

- We began by designing the interval and the data nodes using records and also the tree nodes using a union which consisted of a Node type and a Leaf type.

- We then moved on-to designing a persistent insert function which inserts all the nodes in a persistent manner.

- We also designed a doubly linked list, to hold the root nodes of every tree version.

- We designed a Delete function to delete a node from the latest version of the tree in a persistent manner.

- The insert function was then updated to avoid interval overlaps by placing constraints on the upper and lower bound of each node interval.

- Alternatively we would have had to create a new copy of the entire tree for every new version of the tree, but this would have been more inefficient thus we resorted to creating a copy of every node along the path where a change is to occur. In order to archive persistence.

**(c) Implementation process:**

- In order to carry out our implementation, we relied on the ocamlc compiler, the ocaml command line interpreter, and instructor reviews

- Regularly attending lectures, discussing with and getting clarifications from the teacher and making use of the provided documentation given to us helped deepen our understanding of the ocaml programming language.

- Moving from algorithm to code, required converting the persistent tree data structure provided in CSC 208, with some defined modifications. For testing and debugging of the code, we relied on printed output while running the code.

**(d) General:**

- Some of the information used to accomplish this task was gotten from the Internet and also the resources provided by the teacher .

**PART 3: USE OF PROGRAM.**

**(a) Installation process and use:**

To install this program on a Linux machine:

- Open your terminal.

- Go to the directory in which the program file is found.

- Type ocamlc -c PERSISTENT\_INTERVAL\_TREE.ml and press “ENTER” to compile the program.

- Then ocamlc -o persistent\_interval\_tree PERSISTENT\_INTERVAL\_TREE.cmo. and press “ENTER” to link with object file.

- Now to execute the program, type ./persistent\_interval\_tree and press “ENTER”. This should successfully run the program.

**(b) Demonstration of program functionality:**

• Typical input data and results (output) for specific package operations.

Enter the following commands

1.) I or i -> insert node

2.) P or p -> print list of tree versions

3.) D or d -> to delete node from tree

4.) Q or q -> quit

Enter input : i

Enter node data value : 4

Enter interval lower bound :1

Enter interval upper bound : 2

Enter the following commands

1.) I or i -> insert node

2.) P or p -> print list of tree versions

3.) D or d -> to delete node from tree

4.) Q or q -> quit

Enter input : P

/\*\* List contains the following tree versions \*\*/

Tree version : 1

Data {4}, Interval [1 , 2], Max {2}

•Atypical (boundary) input data and results for specific package operations.

Enter the following commands

1.) I or i -> insert node

2.) P or p -> print list of tree versions

3.) D or d -> to delete node from tree

4.) Q or q -> quit

Enter input : i

Enter node data value : none

-sample error data: “none”.

-Results obtained: Fatal error: exception Failure("int\_of\_string")

-Possible Remedy: The input value must only be an integer and nothing else.

**(c) Program performance:**

The program works by displaying a prompt with commands to perform certain usage operations. For example I to insert, P to print tree versions, D to delete node from tree and Q to quit or exit the program. Selecting any of these commands would cause a particular function associated to it to be executed. This feature makes it easy for end users to operate the software, and also know what to do at any given time.

**PART 4: ADMINISTRATIVE REPORT.**

The group mainly consists of two students who worked collaboratively. Here is a detailed report of each member's contribution to the project:

|  |  |  |
| --- | --- | --- |
| **NAME** | **UB NUMBER** | **DETAILED CONTRIBUTION** |
| DJIMELI KONRAD NIBA | SC14A305 | Implementation of the list\_insert function(Doubly linked list insert),  persist\_insert function, intervalMax function, print\_interval function, last\_list\_elem function, persist\_delete, function tree\_insert function and part of the software documentary. |
| AMATE SUBI YOLANDE | SC13A053 | Defined the interval and the node structures. Worked on the usage function, print\_tree function, and the mainfunc function implementation, program commentary, user documentation and project report. |