A blue and white logo

Description automatically generated with low confidence

6/28/2023

**DSA PROJECT REPORT**

**Group Members:**

Saman Khalid (SP22-BCS-089)

Sara Samreen (SP22-BCS-091)

Syeda Aiman Mumtaz (SP22-BCS-100)

**Implementation:**

AVL and Heap functions performed by Syeda Aiman Mumtaz.

Searching, Sorting and hashing functions performed by Sara Samreen.

BST, Dijakstra and BFS performed by Saman Khalid.

**Explanation:**

The code includes several C++ libraries for input/output, file handling, string manipulation, and data structures.

**BST & AVL**

It defines two structs: CountryData and Node. CountryData represents information about a country, including its name, year, index, and rank. Node is used in the implementation of a binary search tree (BST) and AVL tree.

Several functions are defined for working with binary search trees (BST) and AVL trees, including creating nodes, inserting nodes, searching for nodes, rotating nodes, and printing nodes.

**Dijakstra’s Algorithm**

The code implements the Dijkstra's algorithm for finding the shortest path in a graph. The Dijkstra function takes a cost matrix as input and computes the minimum distances from a given source vertex to all other vertices in the graph.

There are also functions for performing range queries on the tree, deleting nodes, and printing indexes greater than a given value in both BST and AVL tree.

**Sorting and Searching:**

The BST and AVL tree structures provide efficient searching capabilities, and the linked list can be searched by iterating through the nodes. Insertion sorting and Binary Search is implemented in the code.

**Hashing:**

The print (string countryName, int year) function is a form of hashing where the country name and year act as keys to retrieve nodes from the linked list.

**BFS:**

The `BFS` (Breadth-First Search) function performs a level-order traversal of a binary tree or binary search tree. It visits each node of the tree in breadth-first order, starting from the root node and visiting all nodes at each level before moving to the next level. It uses a queue data structure to keep track of the nodes to be processed. In this code, the function prints the `index` values of each node in the tree in BFS order.

**Heap:**

The `heap` class is used to implement a max heap data structure. The `insert` function inserts elements into the heap by maintaining the heap property, where the parent node is greater than its child nodes.

The `printIndexGreater` function creates an instance of the `heap` class and inserts the `index` values of `Node` objects into the heap. It then prints all the elements in the heap that are greater than the given value `d`.