COLLEGE OF BUSINESS, PEACE, LEADERSHIP AND GOVERNANCE

NCSC305 – Parallel and Distributed Computing

Victorino Almeida

ID: 200862

Individual Assignment

There are numerous instances of parallel algorithms, such as:

1. Quicksort
2. Selection sort
3. Insertion sort
4. Counting sort
5. Batcher's Bitonic Sort
6. Radix Sort
7. String Radix Sort
8. Sparse Matrix Multiplication
9. Planar Convex-Hull
10. Three Other Algorithms

These algorithms serve as illustrations for how to evaluate algorithms in terms of their depth and work, as well as how to employ nested data-parallel constructions. Other examples of parallel algorithms include divide-and-conquer methods, parallel rapid sort, and sparse matrix factorization.

Parallel algorithms on sequences and strings, scan (prefix sums), list ranking, sorting, merging, medians, searching, string matching, and other string operations are also included in the NESL library.

CODE:

```
#include <iostream>
#include <vector>
#include <thread>
#include <mutex>
#include <GL/glew.h>
#include <GLFW/glfw3.h>
using namespace std;
// Mutex to protect shared resources
mutex mtx;
// Function to render a single frame
void renderFrame() {
  // Lock the mutex to protect shared resources
  mtx.lock();
  // Clear the screen
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  // Draw a triangle
  glBegin(GL_TRIANGLES);
  glColor3f(1.0f, 0.0f, 0.0f);
  glVertex2f(-0.5f, -0.5f);
  glColor3f(0.0f, 1.0f, 0.0f);
  glVertex2f(0.5f, -0.5f);
  glColor3f(0.0f, 0.0f, 1.0f);
  glVertex2f(0.0f, 0.5f);
  glEnd();
  // Swap the buffers
  glfwSwapBuffers(glfwGetCurrentContext());
  // Unlock the mutex
  mtx.unlock();
}
```

```
// Function to run the main loop
void mainLoop() {
  // Initialize GLFW
  glfwInit();
  // Create a window
  GLFWwindow* window = glfwCreateWindow(640, 480, "OpenGL", NULL, NULL);
  if (window == NULL) {
    cout << "Failed to create window" << endl;
    glfwTerminate();
    return;
  }
  // Make the window the current context
  glfwMakeContextCurrent(window);
  // Initialize GLEW
  glewInit();
  // Set the clear color
  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
  // Create a thread to render each frame
  thread renderThread(renderFrame);
  // Main loop
  while (!glfwWindowShouldClose(window)) {
    // Poll for events
    glfwPollEvents();
    // Render the next frame
    renderThread.join();
```