**National University of Computer & Emerging Sciences Karachi Campus**

**A picture containing graphical user interface

Description automatically generated**

**Implementation and Comparison of A\* Search Algorithm and Dijkstra algorithm with Open Mp and P Threads**

**Project Report**

**Operating System**

**Section: BSE-4B**

**Date: May 26, 2022**

**20K-1042 Saeed Kamran**

**20K-1719 Muhammad Ahsan**

**20K-1720 Abdul Hammad**

**20K-1729 Muhammad Abdullah**

**Introduction:**

Our Project was to Implement the Dijkstra Algorithm and A\* Search Algorithm with OpenMp and Posix Threads. After implementing we have showed that how much time difference is there when the Dijkstra Algorithm is implemented normally, through pthreads and through OpenMp and Same for A\* Search Algorithm.

**Features:**

Our project main feature is that the program shows the time difference when we Implement the Dijkstra Algorithm and A\* Search Algorithm with OpenMp and Posix Threads. Means which algorithm is fast in searching when implemented with the OpenMp and pThreads technique.

**Technology Used:**

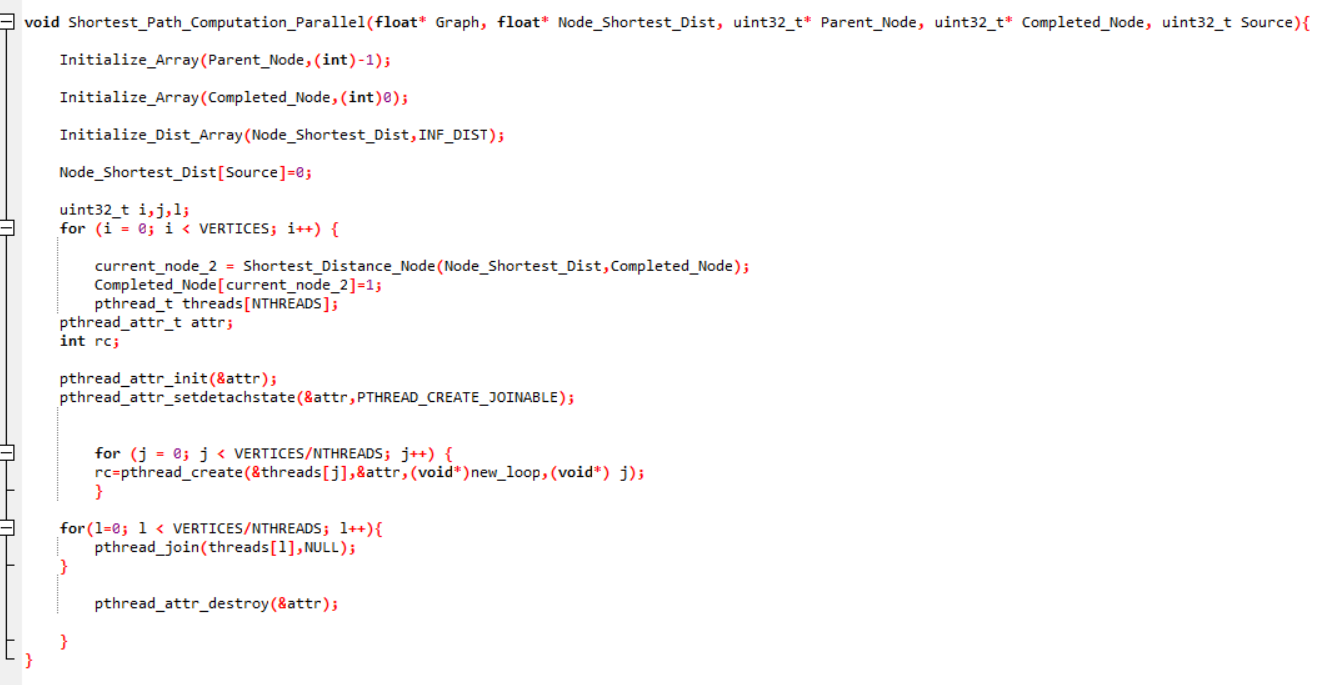
* OpenMp
* Posix Threads
* C and C++ Language

**Code snippets:**

* Dijkstra Algorithm Using pThreads

**Graphical user interface, text

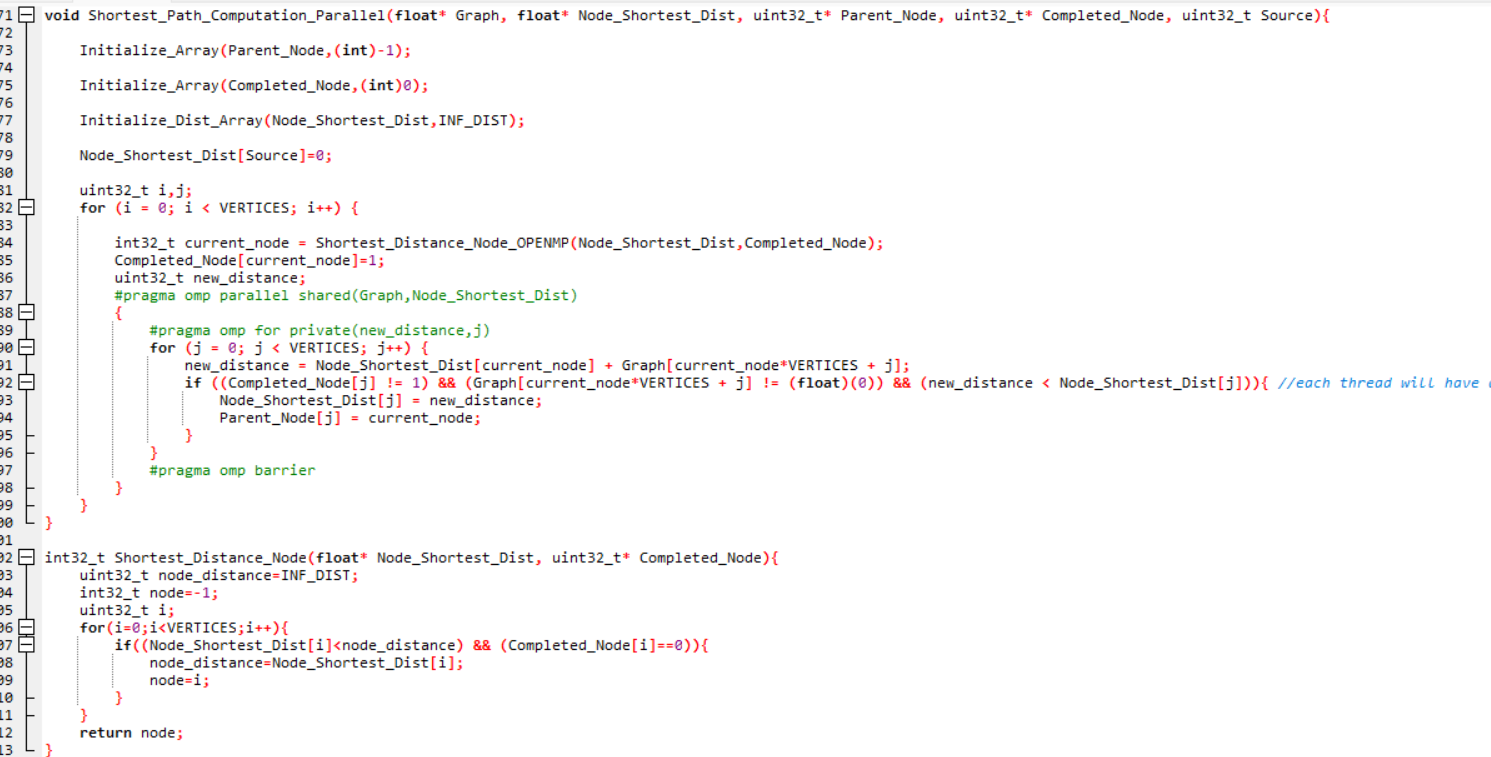
Description automatically generated**

****

* Dijkstra Algorithm Using OpenMp

**Text

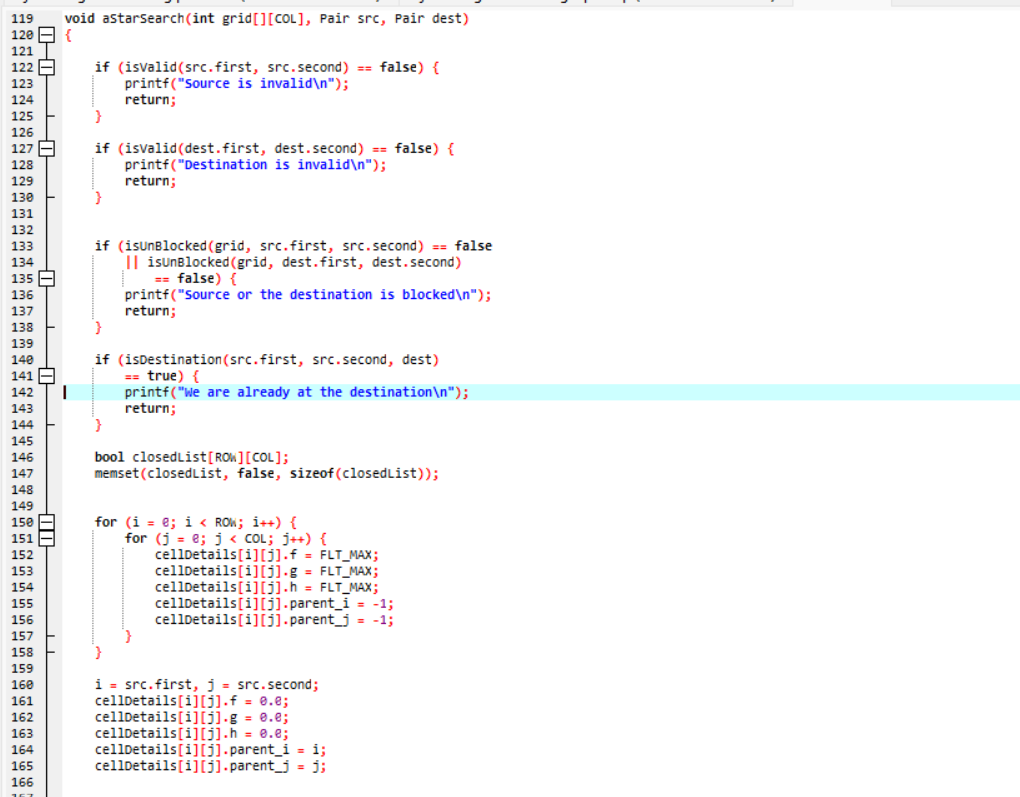
Description automatically generated**

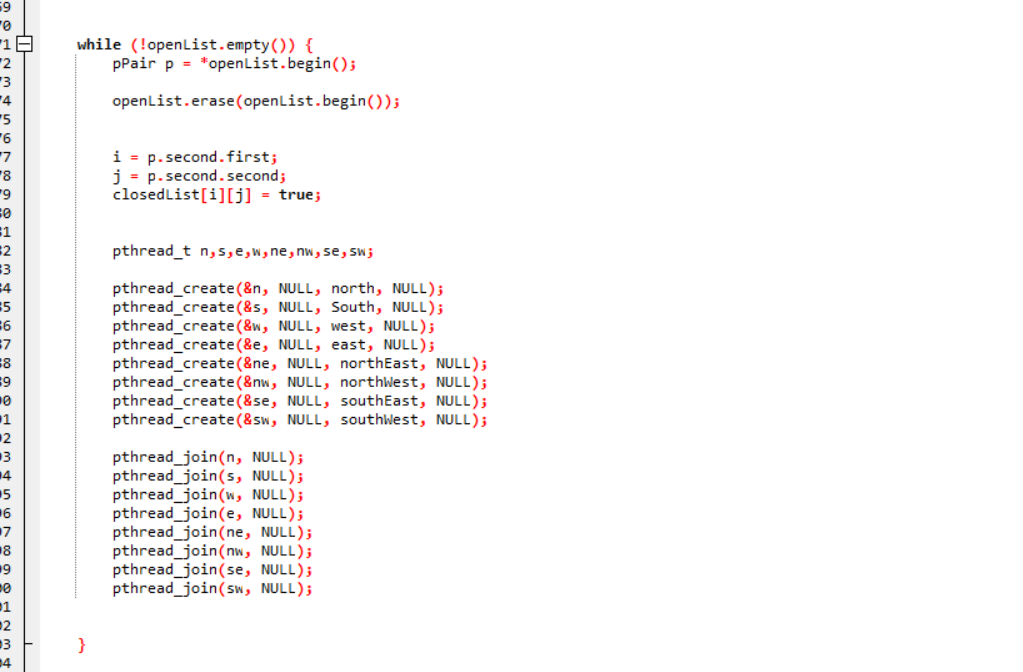
****

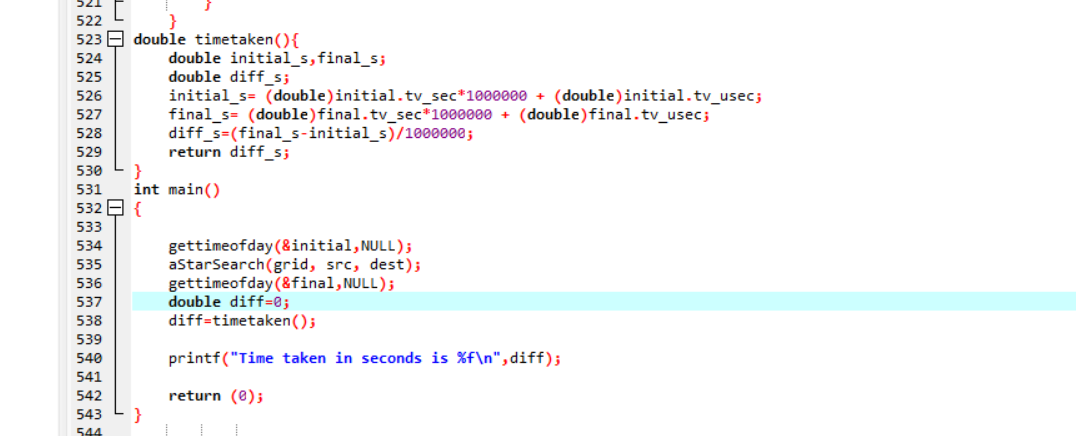
**Graphical user interface, text, application

Description automatically generated**

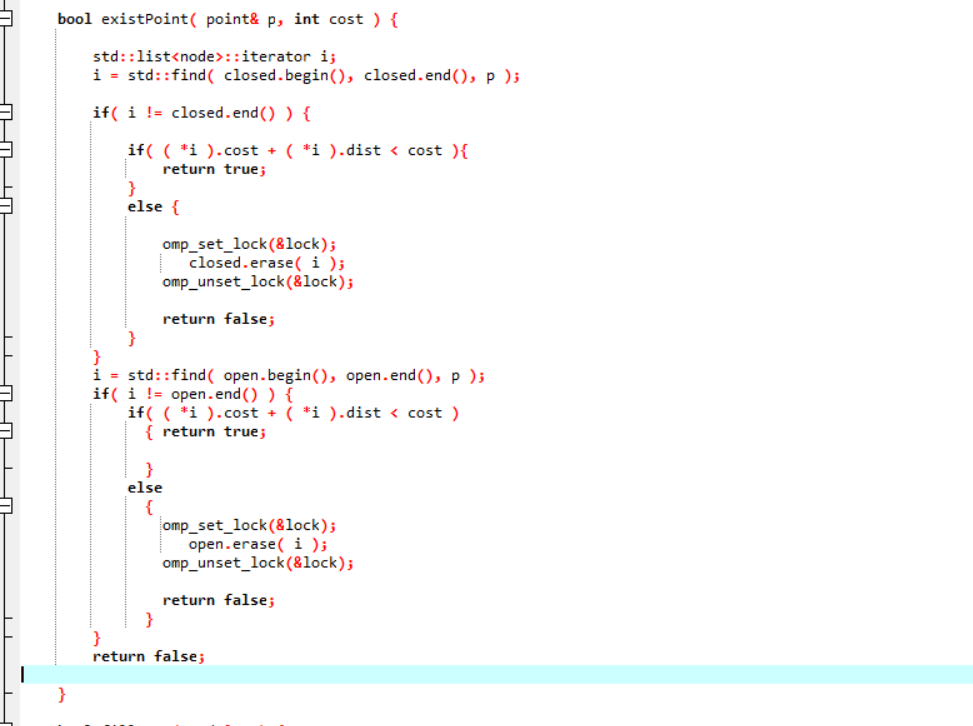
* A\* Search Algorithm Using pThreads

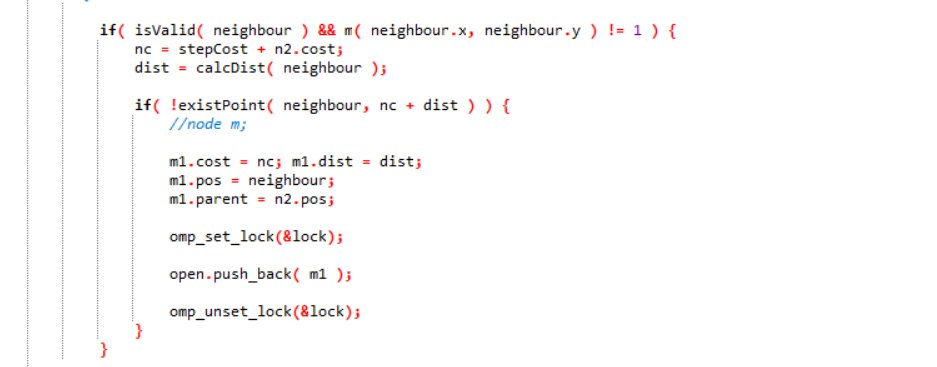
****

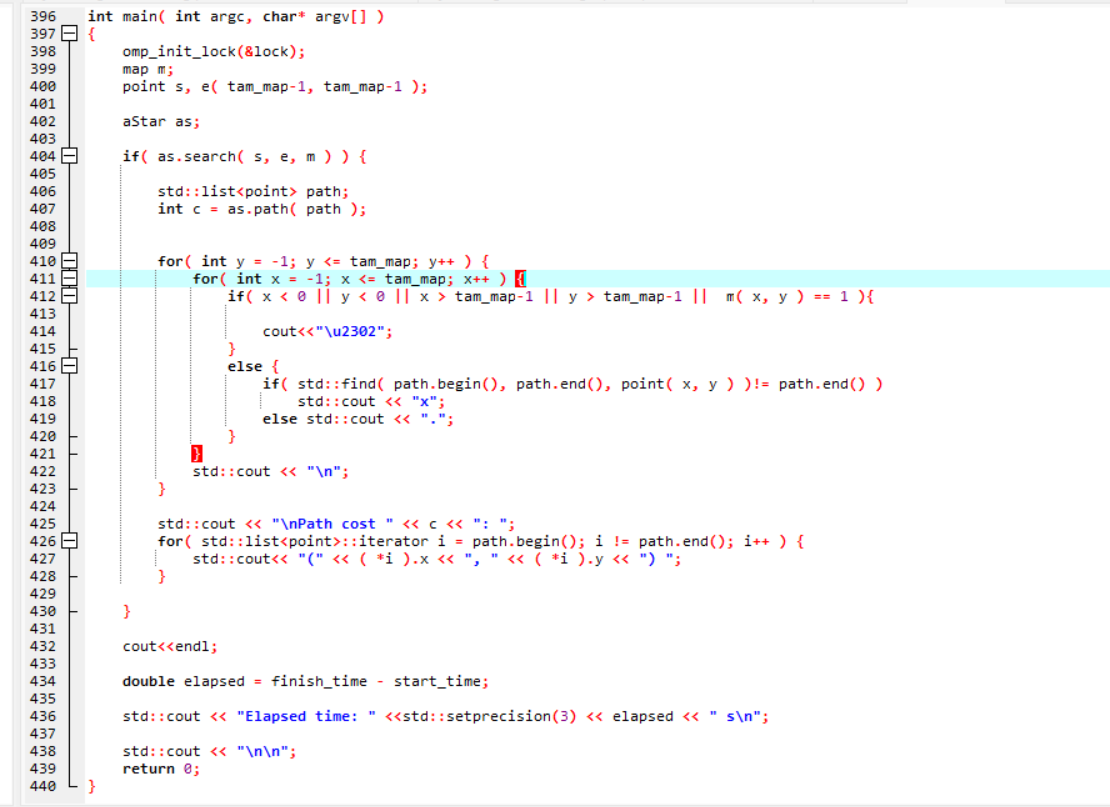
****

****

* A\* Search Algorithm Using OpenMp

****

****

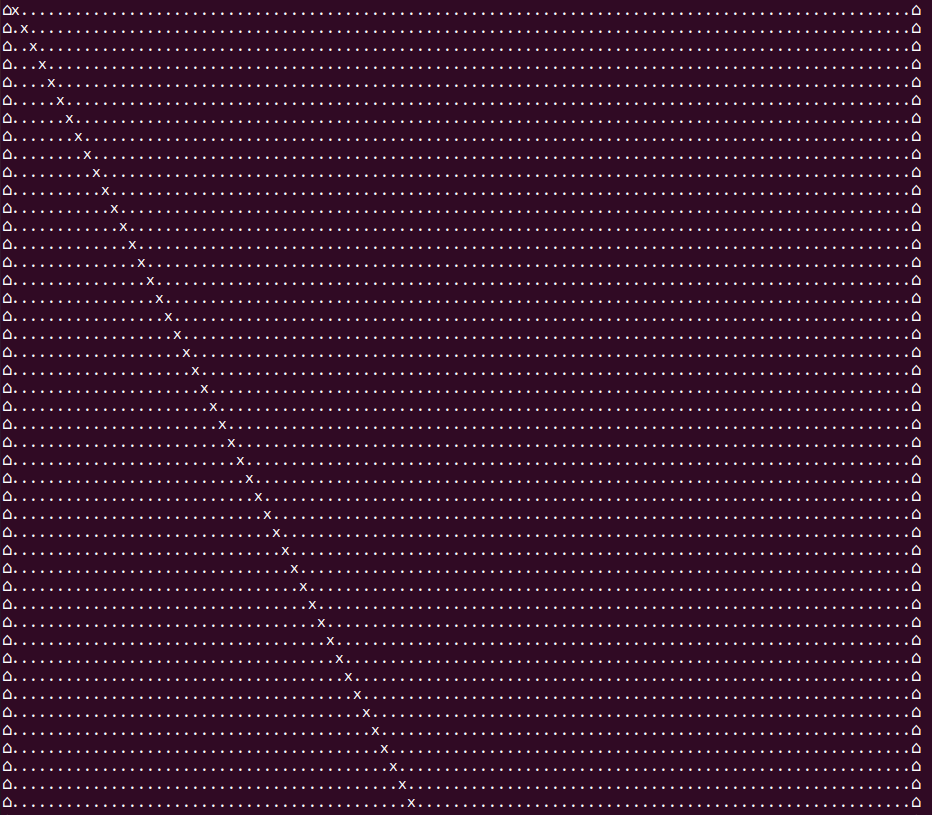
****

**References:**

* Geeks For Geeks
* <https://tuxthink.blogspot.com/2011/02/kernel-thread-creation-1.html>
* <https://riptutorial.com/linux-kernel/topic/10619/creation-and-usage-of-kernel-threads>
* <https://journals.sagepub.com/doi/10.1177/0165551513519212>

**Graphs:**

Graph of A\* Search Algorithm Using OpenMp.



**Conclusion:**

We discovered that using OpenMp and Posix Threads to implement Dijkstra and A\* Search Algorithms resulted in quick computing time as follows:

1. Dijkstra Algorithm with pthreads
2. A\* Search Algorithm with pthreads
3. Dijkstra Algorithm with OpenMp
4. A\* Search Algorithm with OpenMp