**Project Report**

**Serial and Parallel Execution of Searching Algorithms**

**Objective:**

The objective for this project is to run searching algorithms that are already running sequentially and convert their running into parallel execution. The searching algorithms like linear search, exponential search and binary search algorithms will be running over parallel execution. We implemented the execution of the program over sequential and parallel running so we can have a clear image of the difference in time and space complexity affected due to the parallel execution.

**Introduction:**Searching has been a hassle for the entire programming community when it comes to the matter of space and time complexity. The project solves the problem with the huge differences observed in parallel execution of the sequential steps observed in the searching algorithms. However not only this, the project aims to explain the major differences where sequential execution is preferred over the parallel execution however at times the parallel execution is not possible which may cause other problems to rise for the programmers and as well as for the users, so we implement these steps in appropriate and optimized ways.

**Methodology:**

We use OpenMP and multithreading and implement it on different searching algorithms making their executions parallel, comparing their results on large data sets. With the parallel execution we will execute them sequentially and represent their results so the difference can be analyzed easily.

**Graph:**

**Time Complexity Table:**

|  |  |  |
| --- | --- | --- |
| **Searching Algorithms** | **Serial** | **Parallel** |
| Binary Search | 0.002 | 0.003714 |
| Exponential Search | 0.023802 | 0.000488 |
| Fibonacci Search | 0.000046 | 0.000236 |
| Interpolation Search | 0.464 | 0.306 |
| Jump Search | 0.019 | 0.000581 |
| Linear Search | 0.000376 | 0.000432 |

**Applications:**

* Explicitly stored databases.
* Virtual search spaces.
* Sub-structures of a given structure.
* The quantum computers of the future.

**Conclusion:**

Searching is a step-by-step procedure to solve a search-problem in a given search space. A search problem can have three main factors: search space which represents a set of possible solutions, which a system may have, start state from which the agent begins the search and the goal test which observes the current state and returns whether the goal state is achieved or not. Having these search algorithms in parallel saves much time if we have a huge dataset. In this project we have distinguished between sequential and parallel execution of searching algorithms.