LP1 - HPC4

* Title: Parallel Search Algorithm

* Problem Statement:

Design and implement a parallel algorithm utilizing all resources available for -

i) Binary search for sorted array

ii) Best first Search (traversal of a graph to reach a target node in the shortest possible path)

* Objective:

To understand the parallel search algorithm, specifically binary and best first search.

* S/W and H/W requirements:

i) 64 bit processor   iv) Python 3

ii) RAM

iii) Linux Os

* Theory:

I) BINARY SEARCH (Sorted Array)

i) It is a fast search algorithm with a runtime complexity of O(log n)

ii) It works on the principle of divide and conquer. and also requires the array to be sorted.

iii) Binary search looks for a particular item by comparing the middle most item of the collection,

iv) If a match occurs, then the index of the item is returned.

v) If the middle element is greater than the item, then the left sub-array is searchd for the item.

vi) Otherwise, the right sub-array is searched for the item.

vii) For the ordered array (input), and 'x' processors (usually 2), we part our array in $x+1$ parts.

viii) For $k < x$ processors, split the array into n/k groups, and assign a processor to each group, and run binary search on that group.

ix) Thus, the complexity is $O(\log (n/k))$

II Best First search

i) It is an algorithm that traverses a graph to reach a target in the shortest possible path.

ii) Unlike BFS and DFS, Best First search follow an evaluation function to determine which node is the most appropriate to traverse next.

iii) In the parallel formulations of BFS, different processors concurrently expand the nodes in the open list.

iv) However, in this case, the sequential termination criteria fails, and the open list access issues severely limit the performance.

* Steps of Best First Search:

i) start with the Root Node and mark it visited.

ii) find the next appropriate node and mark it visited.

iii) Go to the next level and find the appropriate node and mark it as visited.

iv) Repeat this procedure until the target node is reached.

* Conclusion:

From this assignment, I was able to understand the basics of parallel searching algorithm and hence implement parallel Binary search and Best First search.