Assignment A4 41403

- Title: parallel search algorithm design & implement parallel search algorithm utilizing all available resources for
  - binary search for sorted array
  - best first search (traversal of graph to reach target in shortest possible path.)
- Objectives: To understand the parallel search algorithm specifically binary & best-first search
- outcomes: Understood parallel search algorithm & implemented them successfully

- S/W & H/W: CUDA, g++, 8GB RAM, 64 bit CPU

- Theory:

Binary Search:

- binary search is the fast search algorithm with a runtime complexity of O(log n) It works on the principle of divide & conquer. it also requires a sorted array.
- Binary search looks for a particular item by comparing the middle most item of collection if a match occurs then the index of item is returned.

- If the middle item is greater than the item then the item is searched for in the subarray to the left of middle
- Otherwise the item is searched for in the subarray to the left of the middle item. This continues until the size of the subarray becomes 0.

- For the ordered array that is the input & x processors, we part our array in x+1 parts

- for 1<<x processors split the array into n/k groups & assign a processor to each group & run binary search

the time complexity is thus O(log n/k)

Best First Search

Best first search is an algorithm that traverses a graph to reach a target in the shortest possible path.
Unlike BFS, DFS, Best first search follows an evaluation function to determine which node is most appropriate to traverse next.

- Steps of Best First Search
  - start with the root node, mark it visit
  - find the next appropriate node, mar it visited
  - Go to next level & find the appropriat node mark it visited
  - continue this process until the target is reached.
- In the parallel formulations of BFS, different processors concurrently expand the nodes in the open list. However in this case, the sequential termination criterion fails & the open list access issues severely limit performances

- for Best-first-Search a priority queue is the core data structure. Each processor locks the queue, extracts the best node then unlocks it. Successors of this node are generated their heuristic functions estimated & inserted into the open list (queue) termination is signaled when a solution is found that has better cost than the best heuristic value in the open list

- Conclusion: Successfully implemented parallel binary search & best first search.