Assignment - A3 41403

- Title: parallel sorting algorithm
- problem statement: For bubble sort, & merge sort based on existing sequential algorithms design & implement parallel algorithm utilizing all available resources

- Objectives: Understanding parallel bubble & merge sort.
- outcomes: Understand implemented parallel sorting algorithms

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

- Theory:
  - Bubble sort: there are two phases in this algorithm odd & even phases. n elements are sorted in n phases where n is even.

  - consider a sequence to be sorted $a_1, a_2 \ldots a_n$
  - The odd phase works on the odd indices are compared with their neighbours & are exchanged if found out of order.
  - In a similar fashion, in the even phases, the number at even indices are compared with their neighbours.
  - The sequence is sorted after performing n phases of odd-even exchanges.

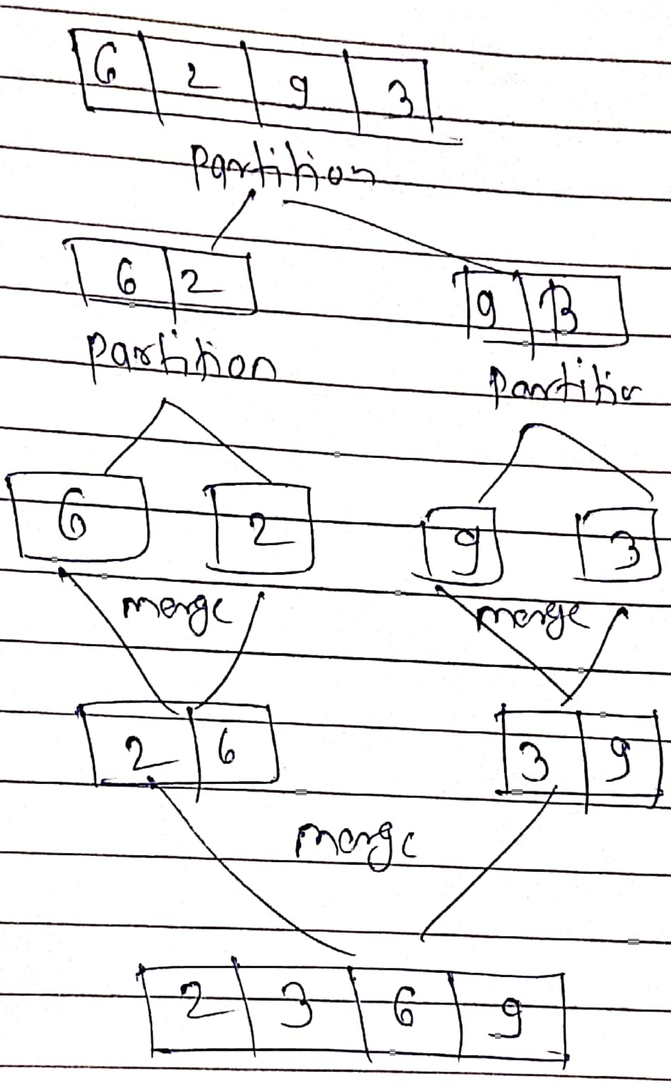
Example.

Step	P0	P1	P2	P3	P4	P5	P6	P7
0	4 ↔	2	7 —	8	5 ↔	1	3 —	6
1	2	4 —	7	8 ↔	1	5 ↔	3	6.
2	2 ↔	4.	7 ↔	1	8 ↔	3	5 —	6
3	2	4 ↔	1	7 ↔	3	8 ↔	5	6.
4	2 ↔	1	4 ↔	3	7 ↔	5	8 ↔	6.
5	~~2~~1 —	2 —	3	4 —	5	7 ↔	6	6.
6	1	2	3 —	4	5 —	6	7 —	8
7	1	2 —	3	4 —	5	6 —	7	8.

— indicates comparison, ↔ exchange

- merge sort first divides the unsorted list in to the smallest possible sublists. compares it with adjacent lists then combines them accordingly.
- It implements parallelism very well by following the divide & conquer algorithm
- It operates in repeated partitions until no more can be achieved followed by repeated compared merges until the original length is achieved.

- Example



- conclusion: successfully understood & implemented bubble & merge sort parallel algorithm.