Assignment - HPC3

- TITLE: Parallel Sorting Algorithm

- PROBLEM STATEMENT: For bubble sort, and merge sort based an existing sequential algorithms, design and implement parallel algorithms utilizing all available resources.

- OBJECTIVES: Understanding parallel bubble & merge sort.

- OUTCOMES: Understood & implemented parallel bubble & merge sort.

- SOFTWARE & HARDWARE REQUIREMENTS: G++, CUDA, Google colab, UNIX OS, 8GB RAM, 64 bit CPU, 128 GB SSD.

- THEORY:

  · Bubble sort: There are 2 phases in this algorithm: odd & even. 'n' elements are stored sorted' in 'n' phases. when n is even

  - Consider a sequence to be sorted <a1,a2...an> The odd phase works on the odd indices are compared with their neighbours.
  - The sequence is sorted after performing n phases of odd-even exchanges
  - Example:

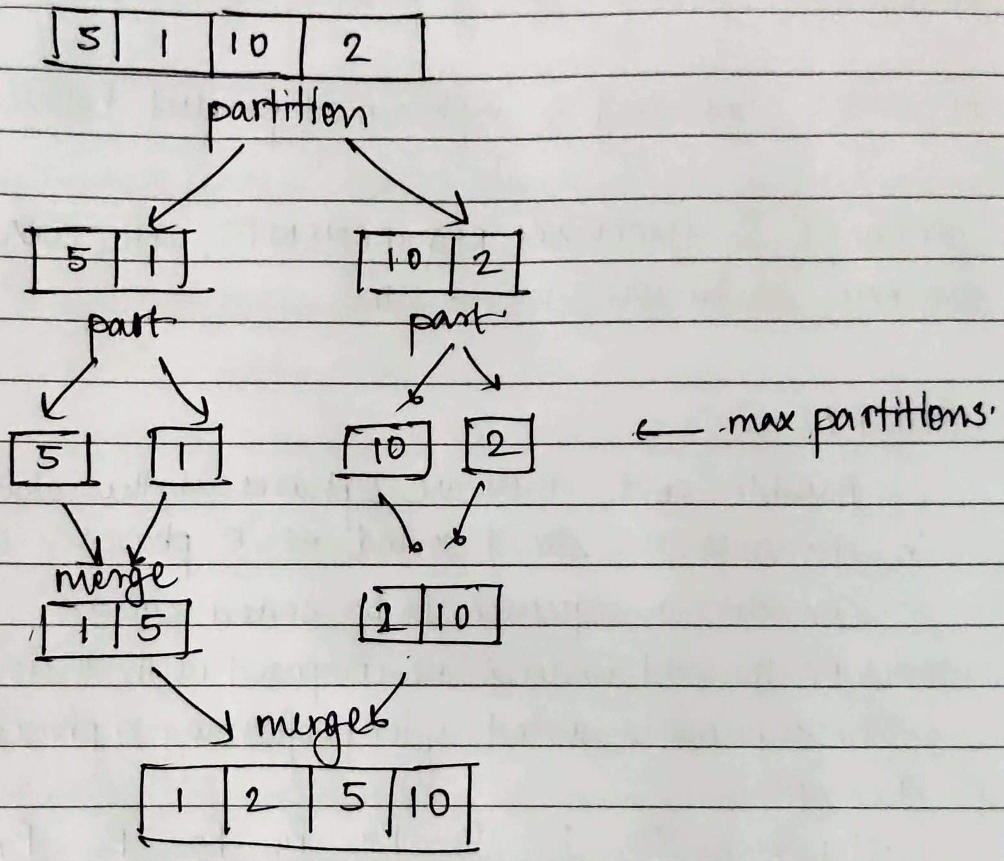
Step ↓	P0	P1	P2	P3	P4	P5	P6	P7
0	4 ↔	2	7 —	8	5 ↔	1	3 —	6
1	2	4 —	7	8 ↔	1	5 ↔	3	6
2	2 —	4	7 ↔	1	8 ↔	3	5 —	6
3	2	4 ↔	1	7 ↔	3	8 ↔	5	6
4	2 ↔	1	4 ↔	3	7 ↔	5	8 ↔	6
5	1	2 →	3	4 —	5	7 ↔	6	8
6	1 —	2	3 —	4	5 —	6	7 —	8
7	1	2 —	3	4 —	5	6 —	7	8

— indicates comparison

↔ exchange.

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

- Example:



* CONCLUSION:

Successfully understood and implemented Bubble and Merge sort parallel algorithm.