

② Merge sort:

merge sort is an example of divide & conquer app.

- In Merge sort unsorted list is divided into N sub-list, each having one element.

- A list of one element is considered sorted to produce sorted list.

12	8	24	17	33
----	---	----	----	----

12	8	24	17	33
----	---	----	----	----

sorted

8	12	17	24	33
---	----	----	----	----

8	12	17	24	33
---	----	----	----	----

8	12	17	24	33
---	----	----	----	----

Merge sort

Que: Explain the Parallel Sorting Algorithms
Bubble and Merge Sort?

⇒ ~~Bubble sort:~~

~~Merge sort:~~

~~Sort the Array in Ascending order~~

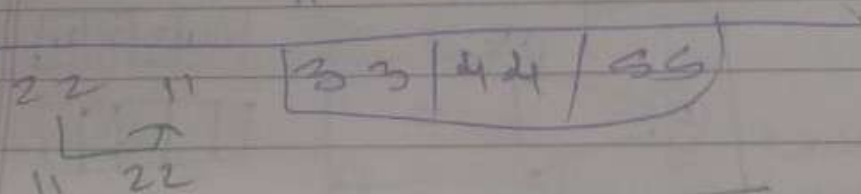
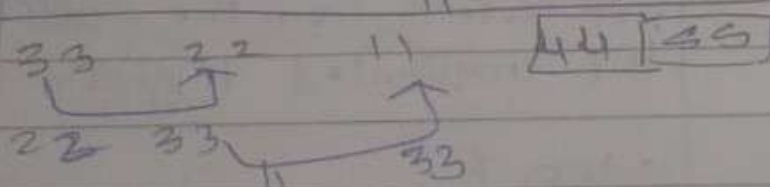
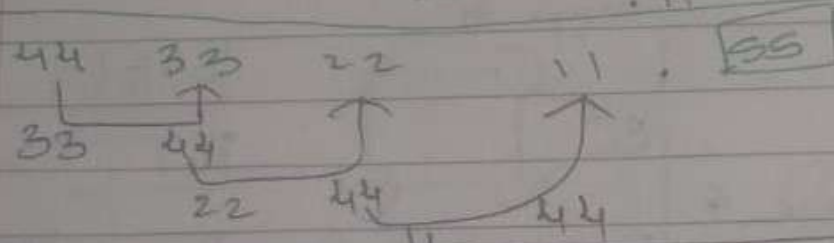
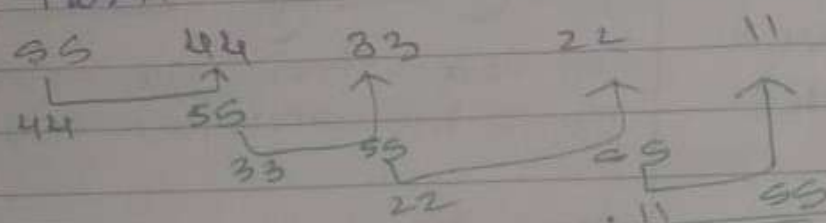
- Bubble sort:

- Sort the Array in Descending order

- compare Adjacent Element

- swap is necessary

- In Every Pass largest Ele get their Actual Pos and Ele is not consider in next Pass.



11 22 33 44 55

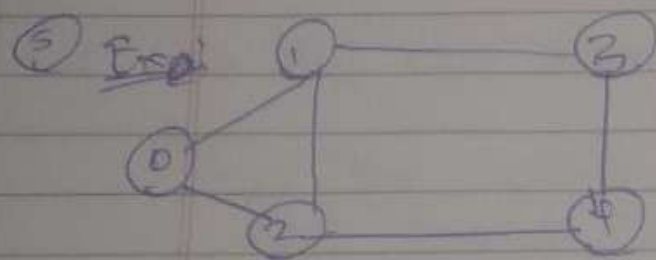
sorted Array are 11 22 33 44 55

6. HPC Applications

Ques: Write short note on Parallel DFS
 Or
 Explain BFS/DFS in Detail.

BFS

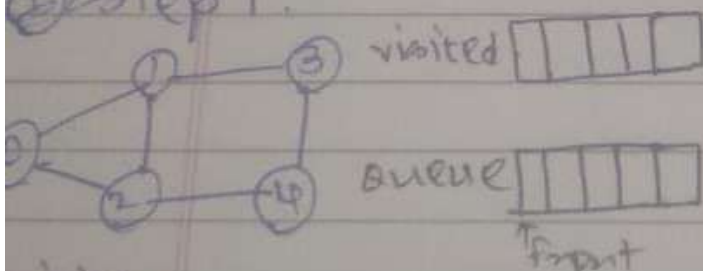
- ① BFS stands for Breadth First Search
- ② BFS use Queue Data structure
- ③ Also known as FIFO search.
- ④ BFS search levelwise in a tree or graph.



- ⑥ Application:
- Finding Shortest Path
 - Peer-to-Peer N/w

Example:

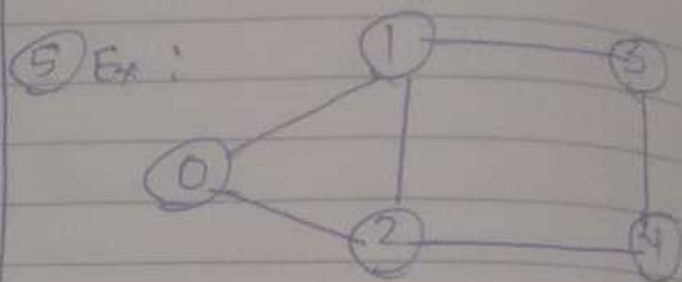
① Step 1:



Initially queue & visited Array are empty.

DFS

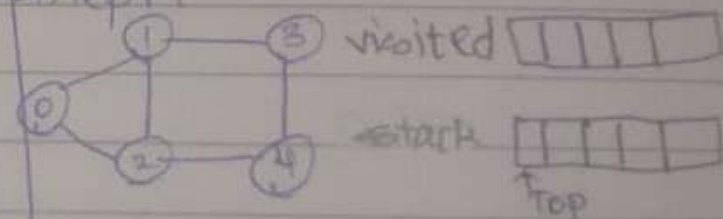
- ① DFS stands for Depth First Search
- ② DFS use stack Data structure
- ③ Also known as LIFO search.
- ④ DFS Traversing Data Root Node to Goal Node



- ⑥ Application:
- Ludo
 - Chess

Example:

Step 1:



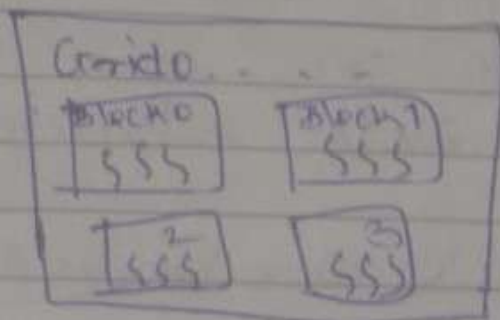
Initially stack & visited array are empty

② launch CUDA kernel:

- launch CUDA kernel by host.

kernel_name << grid, block >>> (argument list)

- There are several grids & the this grids contains no. of blocks. & and this blocks contains no. of threads.



Ques: Explain the procedure to write & launch a CUDA kernel?

=> - CUDA kernel is a function.

① Write CUDA kernel:

~~global~~ - global void kernel_name(argument list)

- host -

- device -

- First one declare global and its return type is void.

- After that write specific kernel_name and write parameters.

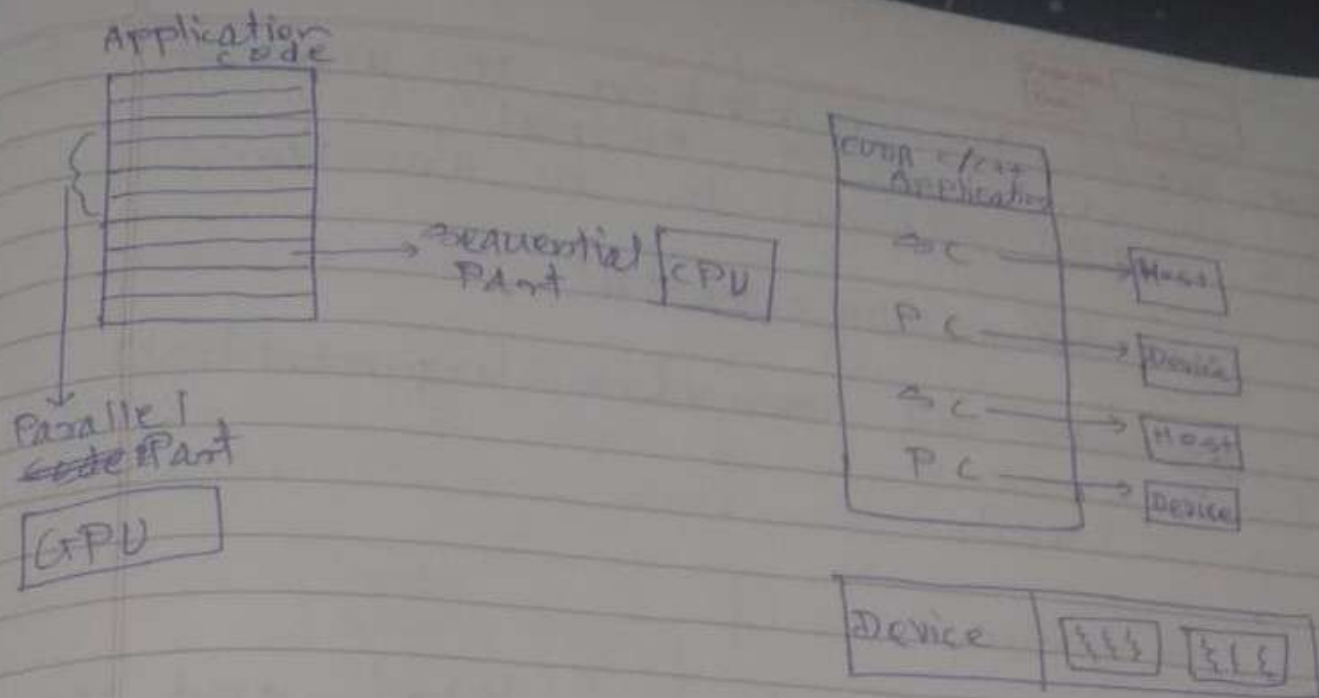
- we can also use host & device

- If the particular function is consider as host then use host

- If the particular function is consider as device then use device

- If the use ~~use~~ host then kernel Access only host memory.

- No support static var & ptr.



Application code contains set of lines. A set of lines can be executed in sequential & parallel Part.

Sequential Part Executed in CPU & Parallel Part Executed in GPU.

In CUDA C/C++ Application Randomly Executed in sequential code and parallel code.

Sequential code Executed in Host

Parallel code Executed in Device

Device contain no. of blocks.

Blocks contain no. of threads.

Thread Executed Parallel code.

- CU :- CU stands for Control Unit.
- CU control the information.

- ALU :- ALU stands for Arithmetic Logical Unit.
- ALU performs Arithmetic operations.
- ~~ALU~~ performs comparison operations.
- load the data into ALU.

- DRAM :- DRAM store info in longer time.
- DRAM is slower.

- cache :- store temporary data.

GPD :-

Write about parallel programming in CUDA-C++
Que: Explain in detail CUDA Programming Model.
=> - CUDA stands for Compute Unified Device Architecture.

- CUDA is a library (Package) developed by Nvidia.

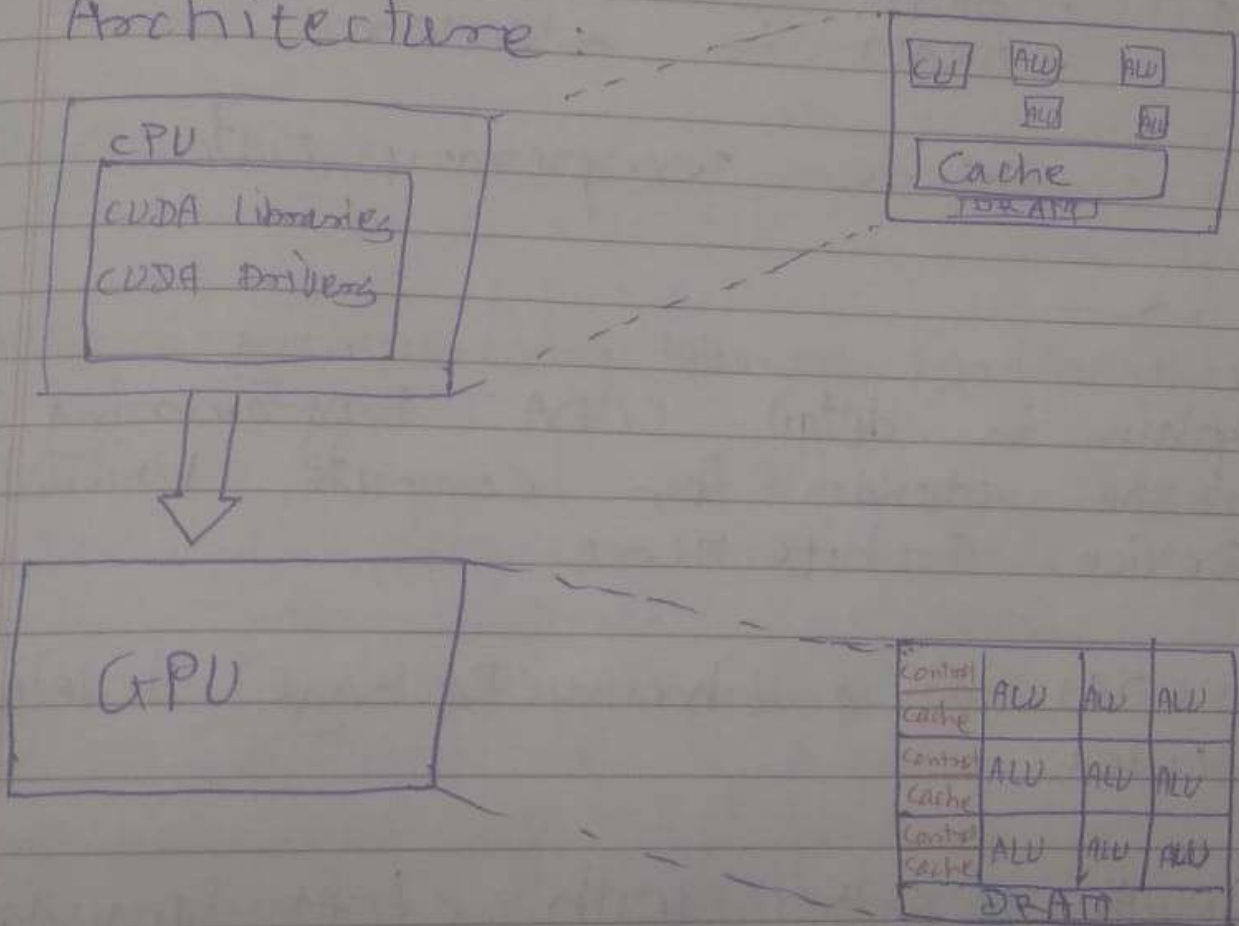
- CUDA works with C/C++ language.

- CUDA is parallel computing platform.

Ques: What is CUDA? / CUDAC?

- - CUDA stands for Compute Unified Device Architecture.
- CUDA is a library (Package) develop by Nvidia.
- CUDA works with c/c++ language.
- CUDA is Parallel computing Platform

Architecture:



1. - CPU stands for Central Processing Unit.

- CPU required More Memory than GPU is slower than GPU

- CPU Not suitable for Parallel Instruction Computing.

- CPU contains CUDA Libraries, CUDA Drivers

- CU :- CU stands for controlling
- CU control the information.

- ALU :- ALU stands for Arithmetic
logical Unit.

- ALU perform Arithmetic
~~each as~~ comparison of
logical operations.

- load the data into ALU.

- DRAM :- DRAM store info in Lo

- DRAM is slower.

- cache :- store Temporary data.

GPU :- ~~~~~

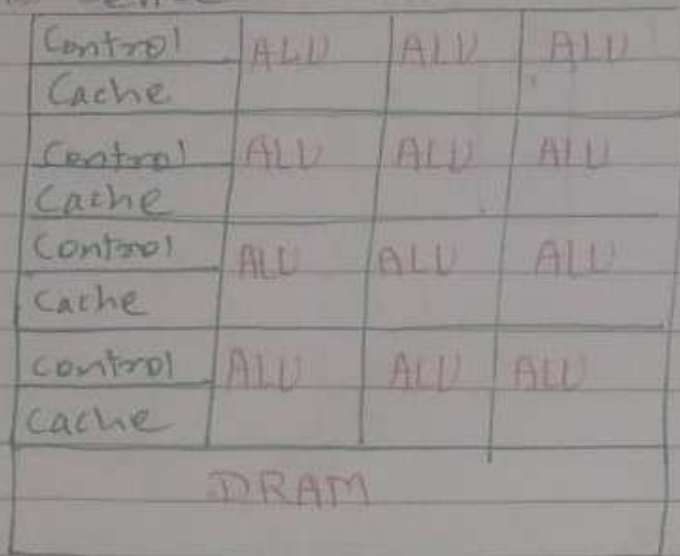
- CPU :- CPU stands for Central Processing Unit.
- CPU required More Memory than GPU
 - CPU is slower than GPU
 - CPU Not suitable for Parallel computing.
 - CPU contains CUDA Libraries, CUDA Driver
 - In CPU there are several parts:
 - CPU is most - small data size.

5. CUDA Architecture

Ques: What is GPU Architecture explain in detail with diagram?

⇒

- GPU stands for Graphical Processing Unit
- GPU required less Memory than CPU
- GPU is Faster than CPU (Central Processing Unit)
- GPU suitable for Parallel Instruction Processing
- GPU is Device - Huge Amount of Data.



- ALU :
- ALU stands for Arithmetic Logical Unit
 - ALU Performs Arithmetic operations, ~~math~~ comparison oper^{ns}, logical operations.
 - Load the data into ALU.

AM :

DRAM store information in longer time

DRAM is slower

Reg : - store ^{temporary} ~~Temporary~~ data