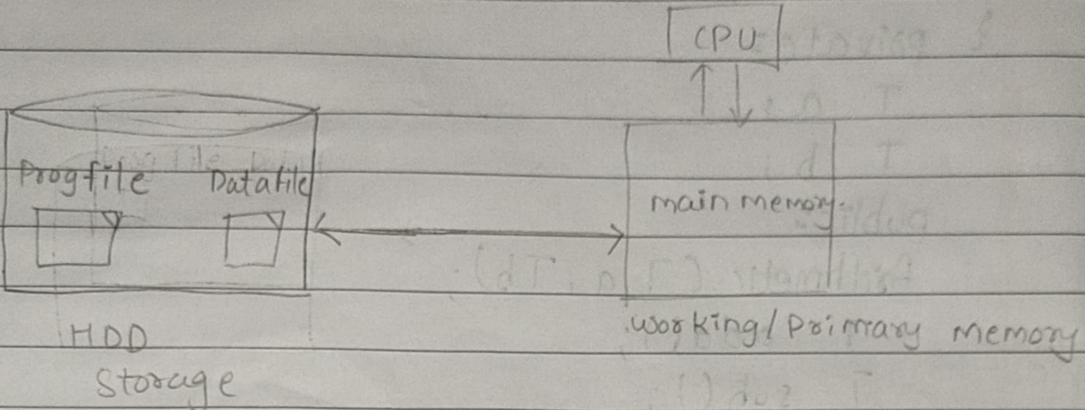
Date
Page 20/12/21

Introduction ~~to~~ Data structure :

- Data Structure :- Data structure is collection of data element so that operation can be done on that data efficiently inside main memory.



- Data base :- Large commercial data is store in HDD storage in the form of table this arangement in permanent storage is Database.

- Commercial Data → operational → Daily used data
  → legacy or old data → old data not used frequently

- legacy data is store in an array of disk called data warehouse and the algorithm for ~~analyzig~~ analyzing the data is data mining algorithm

- Big data → Analyzing large and large amount of data on internet and ~~studyigabouth~~ studying about that is known as big data.

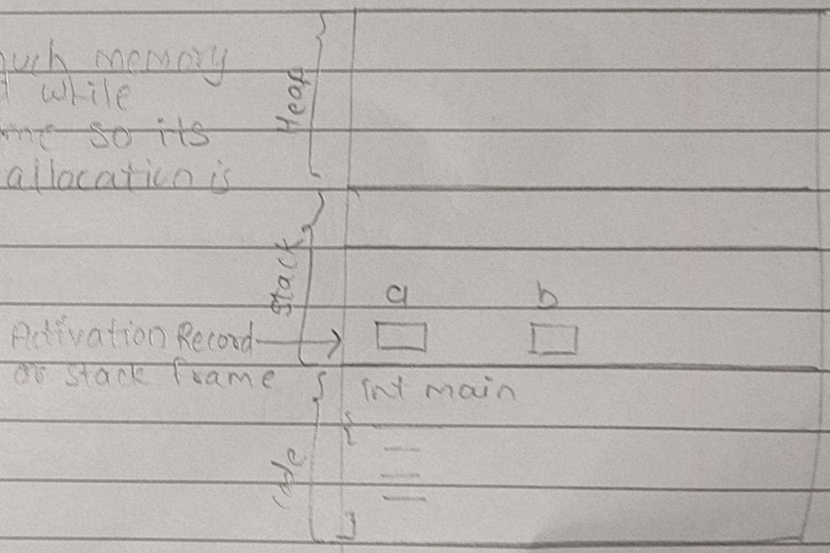
Static vs heap Memory

Main Memory (4 GB) is divided into huge number of segment of memory (64 KB)

∴ 1 segment of memory (64 KB).

* Static Memory → whose memory is defined while compile time or before runtime is static.
∴ The memory which is allocated during compile time is known as Static Memory allocation.

Eg:) int main
{
  int a, b
  ↑ How much memory is required while compile time so its memory allocation is static



Stack	Heap
i) Can be directly accessed	i) Cannot be directly accessed
ii) Stack of memory	ii) Pile of memory
iii) Organized	iii) Unorganized
iv) Is automatically created and destroyed	iv) Neither it is automatically created nor destroyed

Eg) Allocating memory in heap i.e Heap memory allocation

```
C++ → int * p
      p= new int[5]

C → int * p
    p= (int *) malloc( size of(int))
```

Deallocating Memory

```
delete []p;
~~delete ;~~ p=NULL;
```

Types of Data Structure.

→ Physical Data Structure

- ✱ Fixed size → Array
- ✱ Linked List → variable in size (always created in Heap)

→✱ Logical Data Structure.

1) Stack (LIFO)] linear
2) Queue (FIFO)] linear

3) Trees] Non-linear
4) Graphs] Non-linear

5) Hash table] - linear / Tabular

logical Data Structure are implemented on physical Data Structure.

* ADT(Abstract Data type)

Data type $\xrightarrow{\text{Representation}}$ int
$\xrightarrow{\text{Operation}}$ Arithmetic, logical, increment & Decrement

Abstract Data type :- Abstract Data type is hiding the complexity of some operation. ~~ie~~ ~~co~~ on a Data type i.e suppose we want to perform addition (operation) of two int(Data type) we dont want to know how compiler actually adds those integers we just wants them to be added is known as Abstract datatype wherein the operation & datatype are not explained on how it is actually done.

Date ....
Page ....

Time and space complexity.

→ Time complexity is the computational complexity it takes to run an algorithm

```
for (i=0; i<n; i++
{
    =
    -
}
```

Here time complexity is n or you can also say that order of n or O(n)

Example

```
void Add (int n
{ int i,j;
  for (i=0; i<n; i++)  → n+1
  {
    for(j=0; j<n; j++) → n(n+1)
     { c[i][j] = A[i][j] + B[i][j] → n×n
     }
   }
 }
```

Time function $f(x) = 2x^2 + 2x + 1$

Order of	n^2	$\rightarrow O(n^2)$
Big O of	n^2	$\rightarrow O(n^2)$
tetha of	n^2	$\rightarrow \theta(n^2)$
Omega of	n^2	$\rightarrow \Omega(n^2)$