Allocating sectors/blocks to Files in HDD

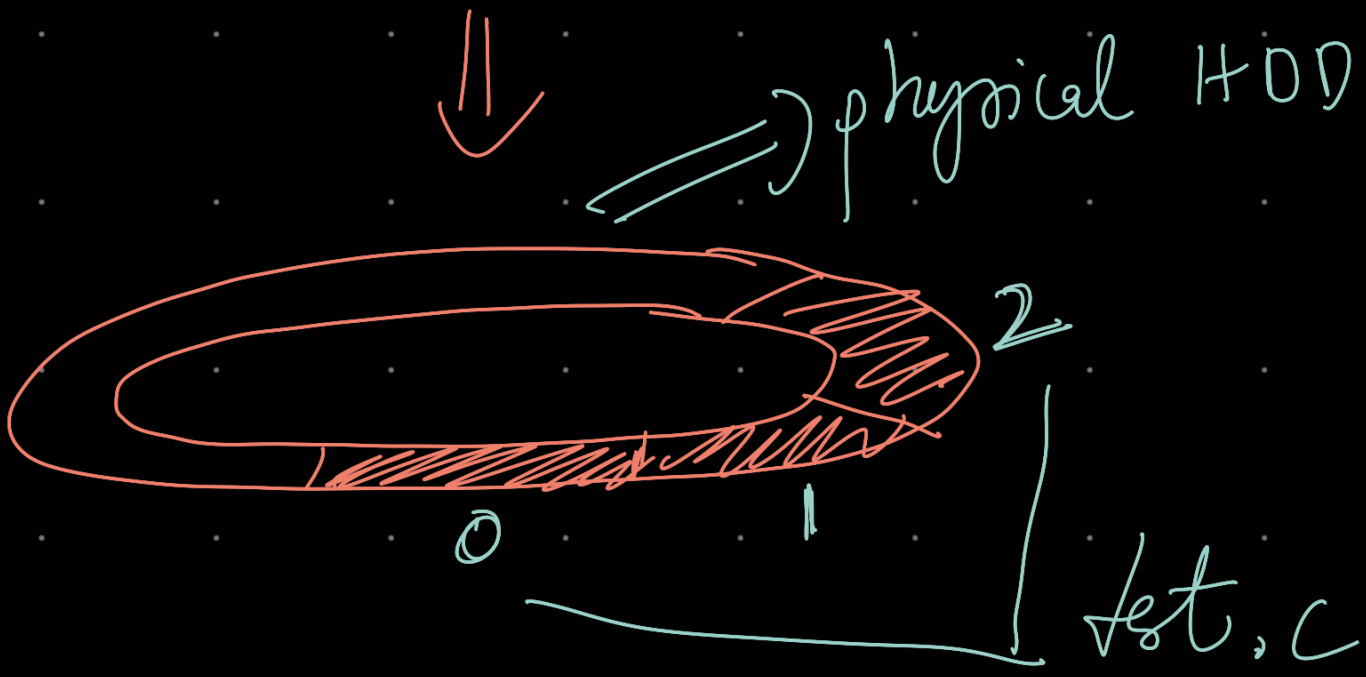
* Disk has direct access nature and each sector can be accessed via R/W head

① Continuous Allocation

file name	start	length
test.c	0	2
img.jpg	14	5
⋮	⋮	⋮

→ starting block no or sector no.

(Directory table)



* Access time / seek time ↓ less

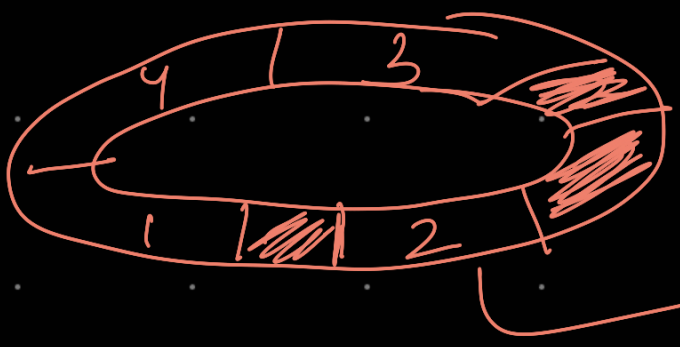
Problems

1. Internal fragmentation

File → 3.5 blocks

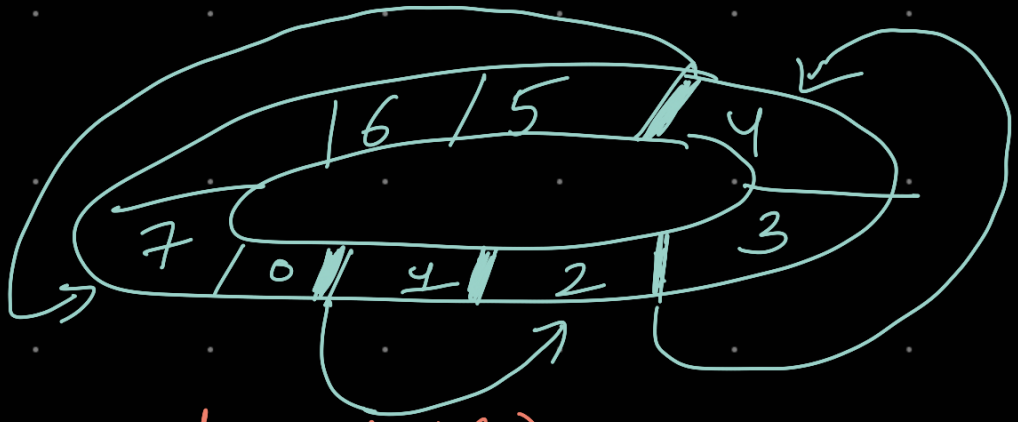
↳ Have to allocate 4 blocks, last block half wasted

2. External fragmentation



4 block empty but can't allocate to a file as not continous

(2) Non-Contigous / Linked



(Directory table)

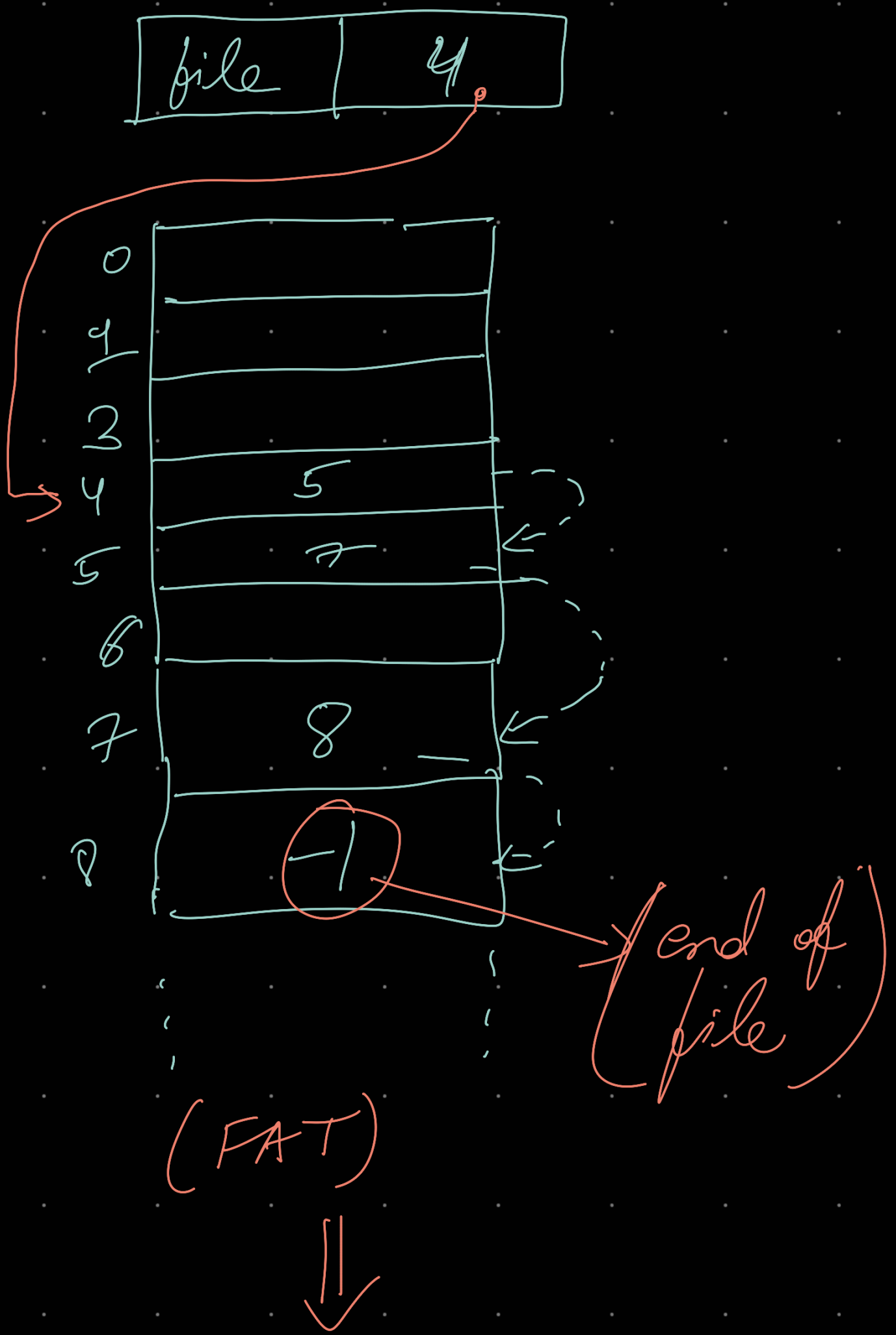
file	start	end
fst.c	0	7

* No external fragmentation
* Internal fragmentation is there.
* Some bytes reserved for pointer
  - Eg → 512 B → Block size
    Pointer → 4 B
    Data → 512 − 4 → 508 bytes
* **FAT is type of non-contigous**

FAT → File allocation table

* At beginning of each partion store FAT table.
* Directory entry will contain file name and start index in

FAT table.



↳ physical storage of file

* Space wasted for FAT storage

E.g → HDD → 20 GB

Block size → 1 KB

No. of blocks → 20 Million

FAT size → 20 Mill * size of each entry (4 bytes)

↓

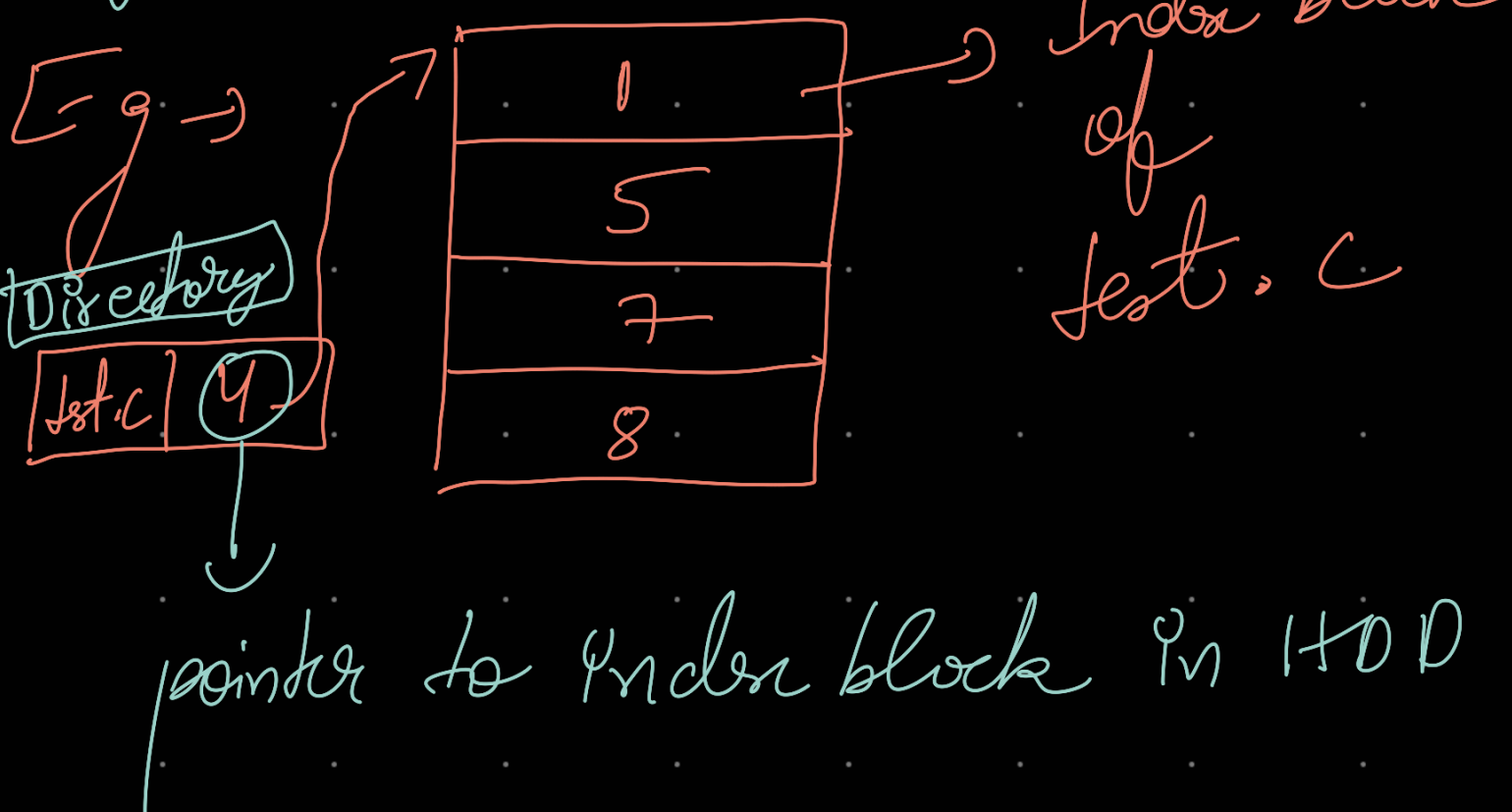
80 MB wasted

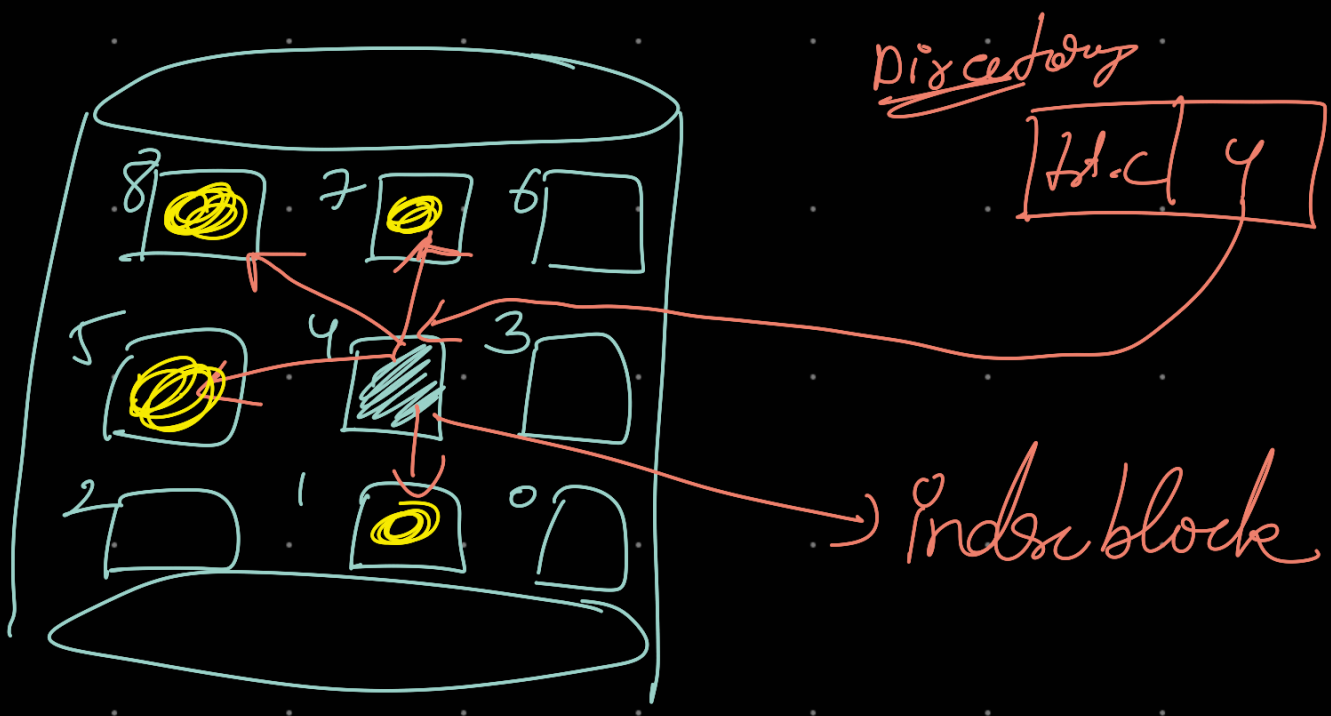
* FAT is loaded into RAM for

· optimization

③ Indexed Allocation

* Each file has its index block which contains pointers to all physical block occupied by that file





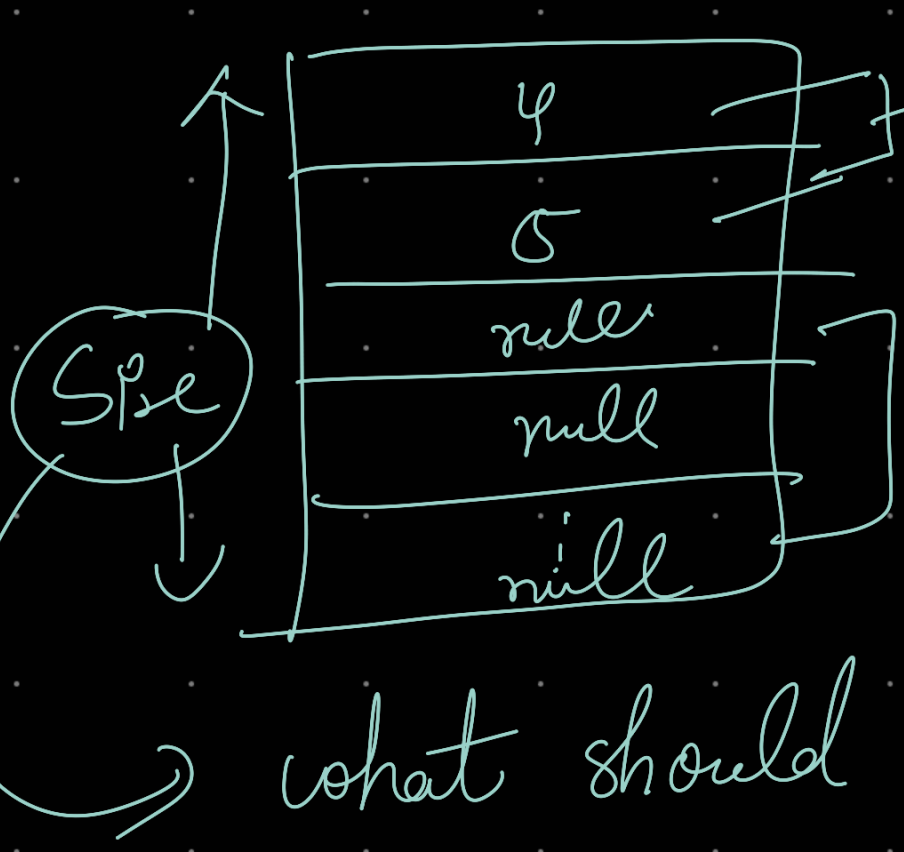
So. → For 10,000 file

⇓

10,000 Index block

Here question arise, what should be the size of index block?

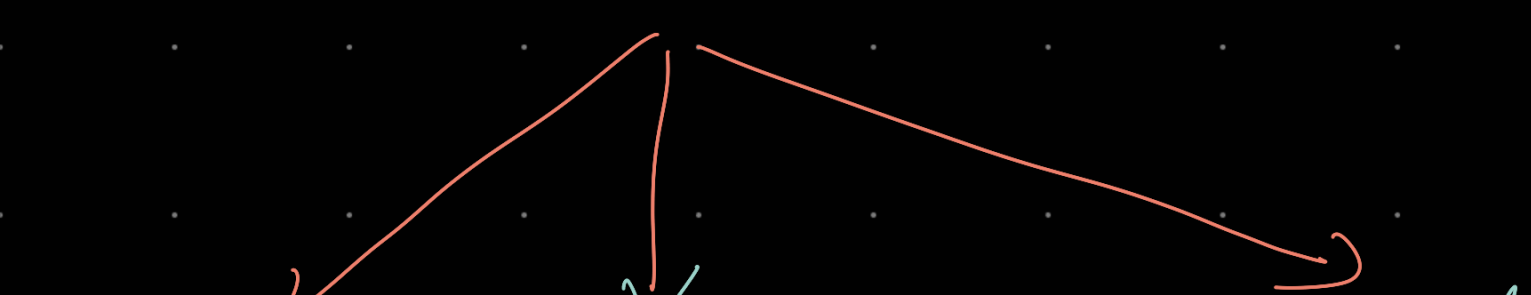
* Let say a file occupies 1 or 2 blocks



what should be ideal size to minimize wastage

* If block size is very small it may not be able to hold enough pointers for very large file consider a video file of 5 GB.

To solve this issue



Linked Schema | Multi-level Index | Combined Schema

* Index block is normally 1 disk block

E.g → Size of disk = 500 GB

Size of 1 block/sector = 512 KB

Total blocks = 1048576000

or sectors

Total bits required ⇒ $\log_2 N$

to represent

≈ 30 bits

* 30 bits can represent all the sectors with unique number

⇓

Now, file → 5 GB

Sectors required ⇒ $\frac{5\,GB}{512\,KB}$

$= 10,240$ sectors

Total pointer size $= \frac{10,240 * 30 \text{ bits}}{1 * 1024}$

$= 300\,KB$ → Size of index block

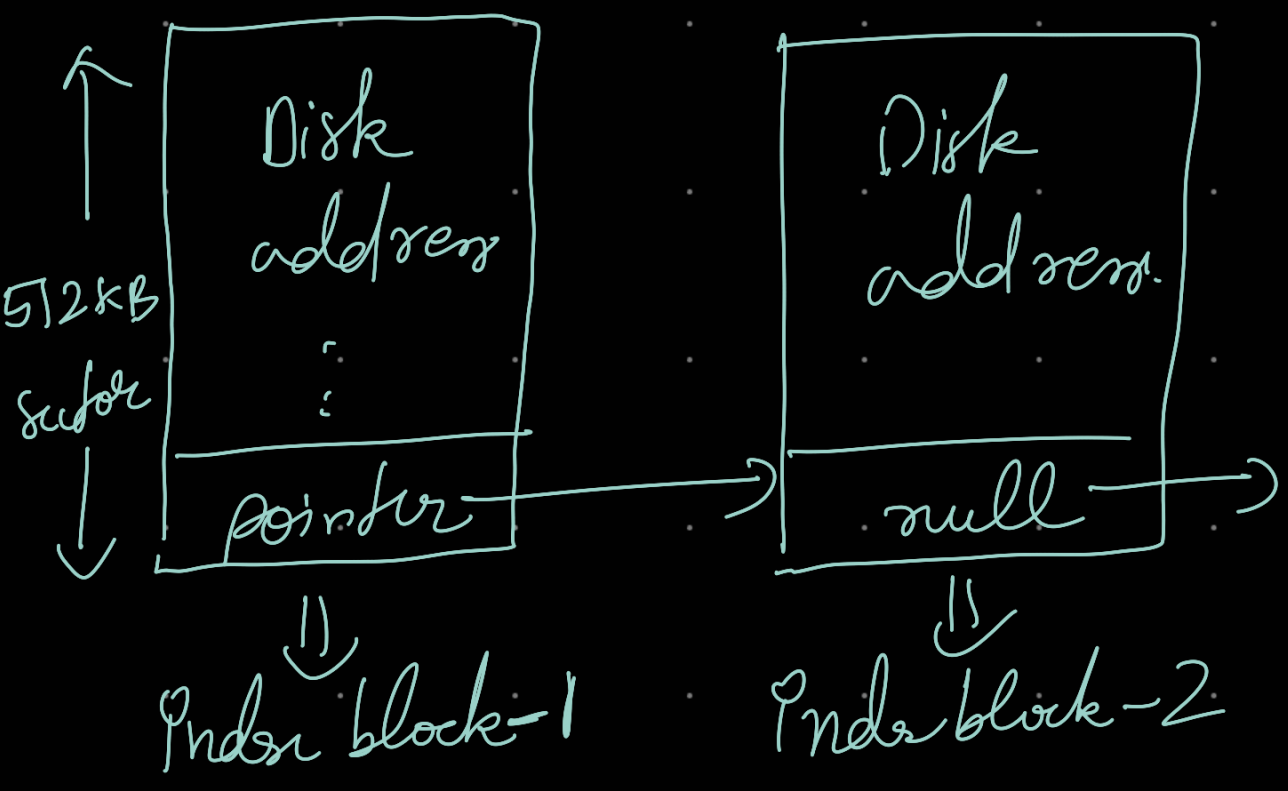
This can be fit in 1 block of HDD

* Now Assume 1000KB

⇓

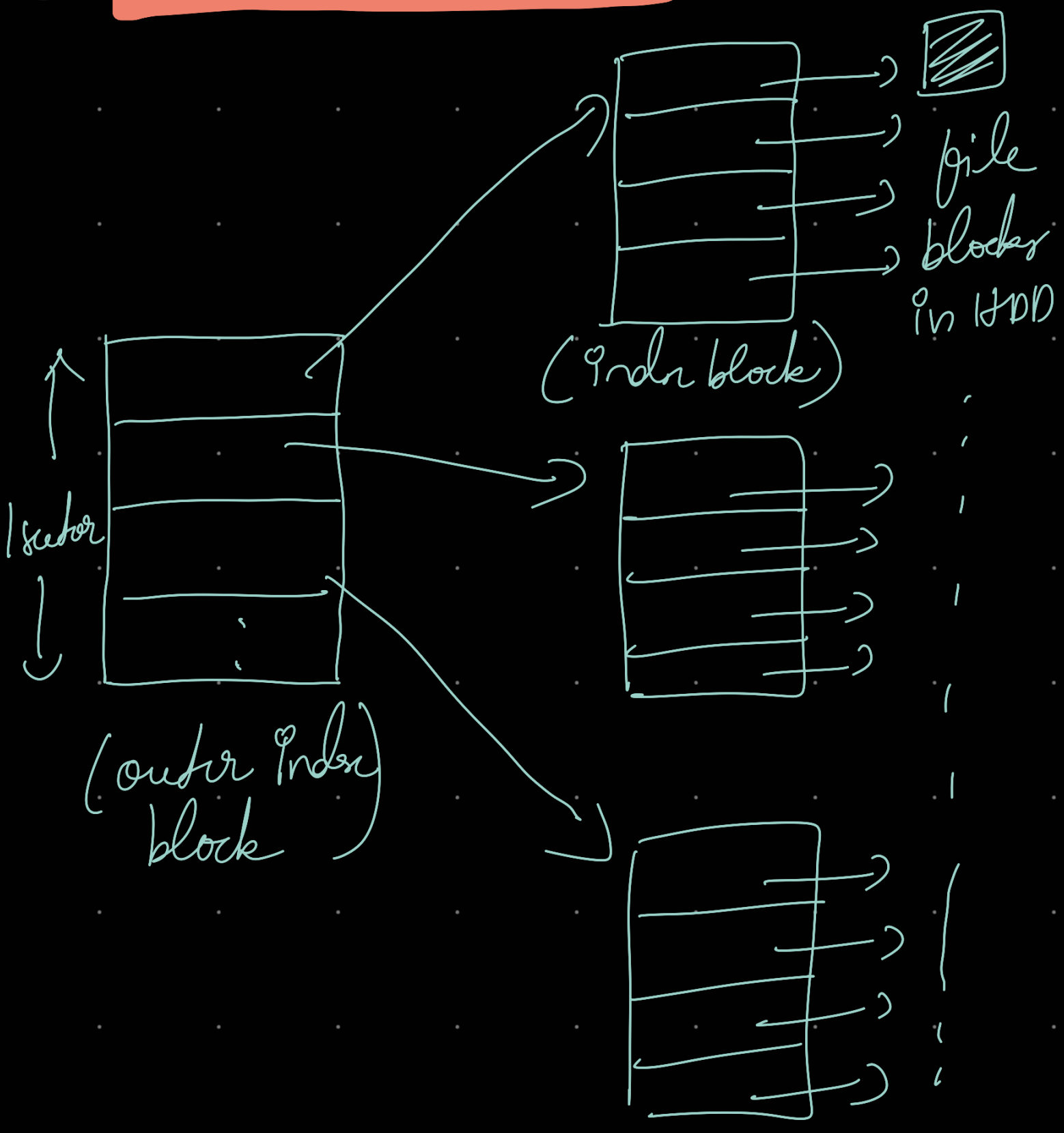
can't be fit in 1 HDD block

① Linked Schema

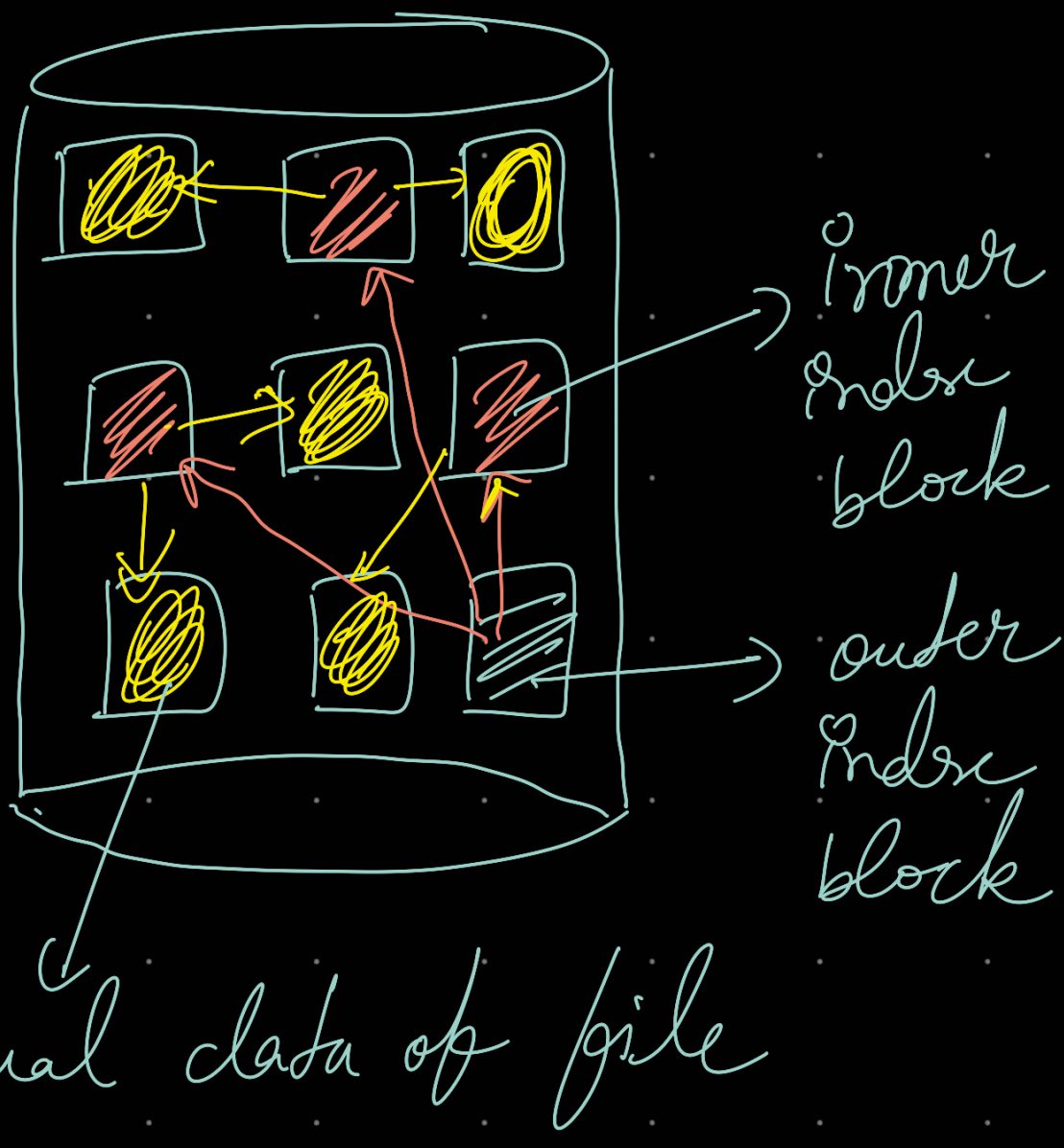


* 1 Pntr block = 1 sector in HDD

(2) Multi level index



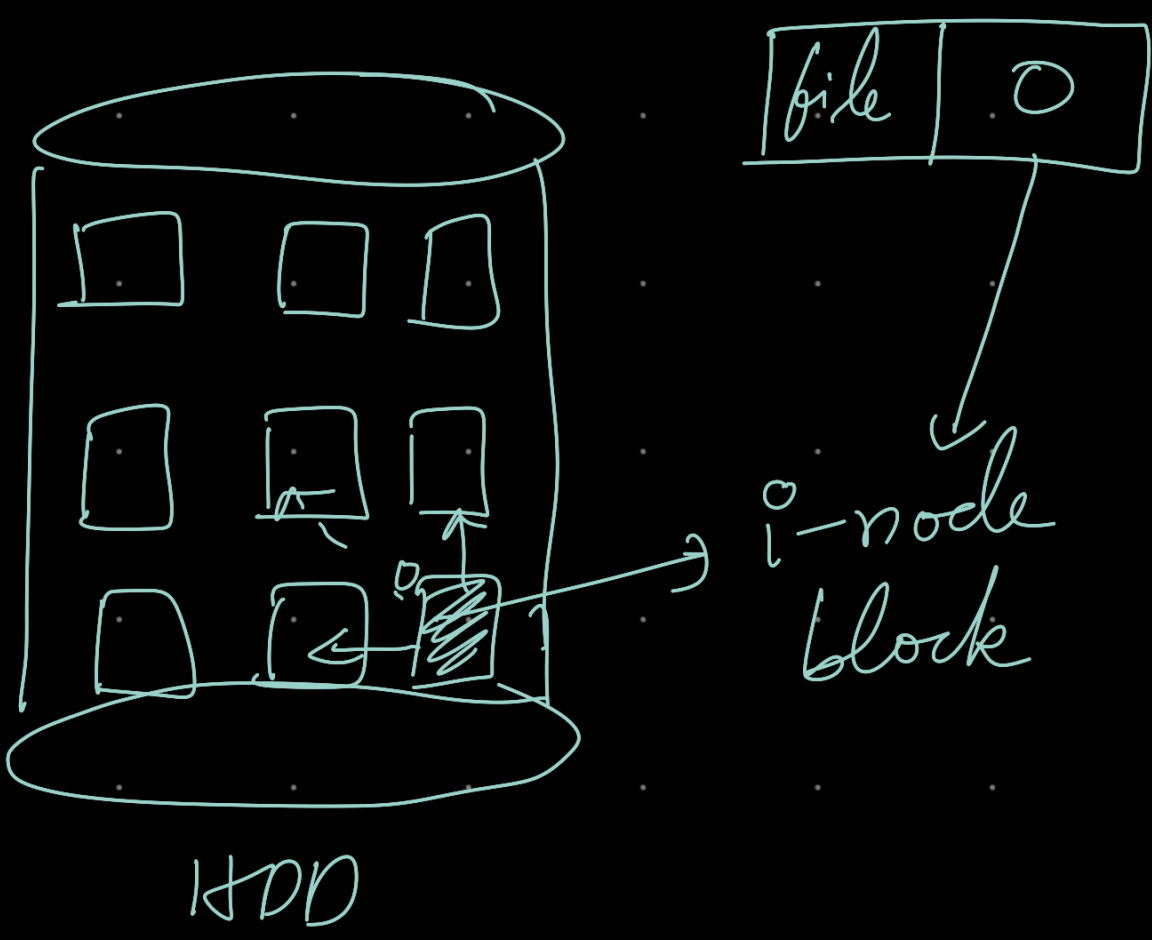
HDD view

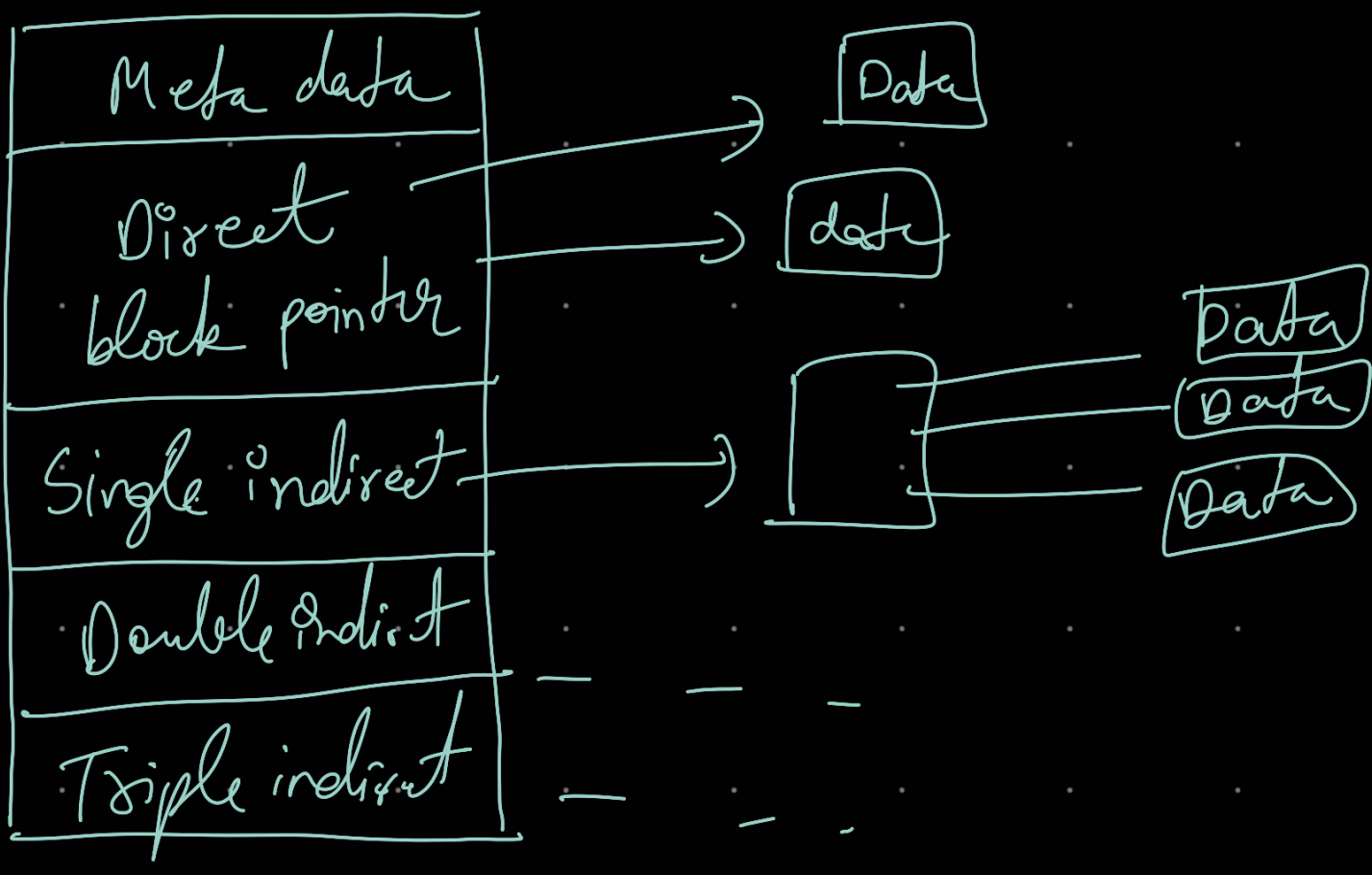


* It can be upto N-level
* Directory contains address of outer index block only

③ Combined Schema

- E.g Unix i-node structure
- Contains both linked and multi-level
- Directory contains block address of i-node in HDD





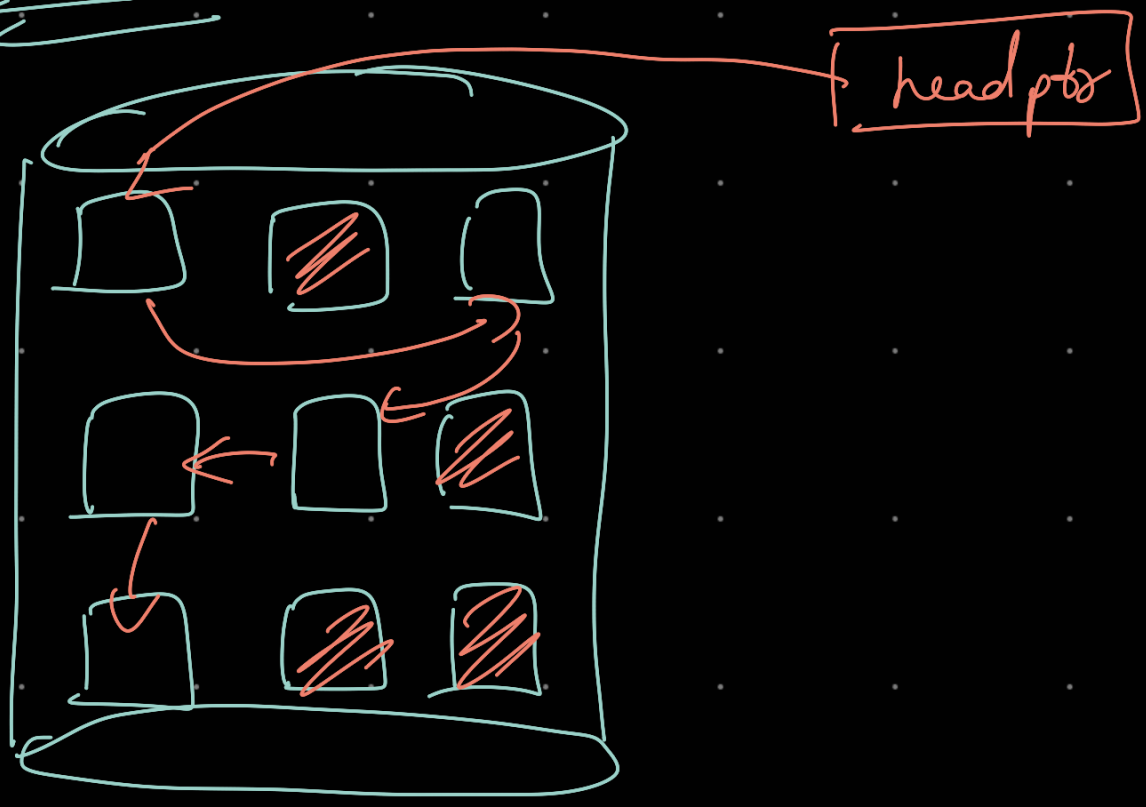
Free space management

* Free blocks can be tracked using two ways

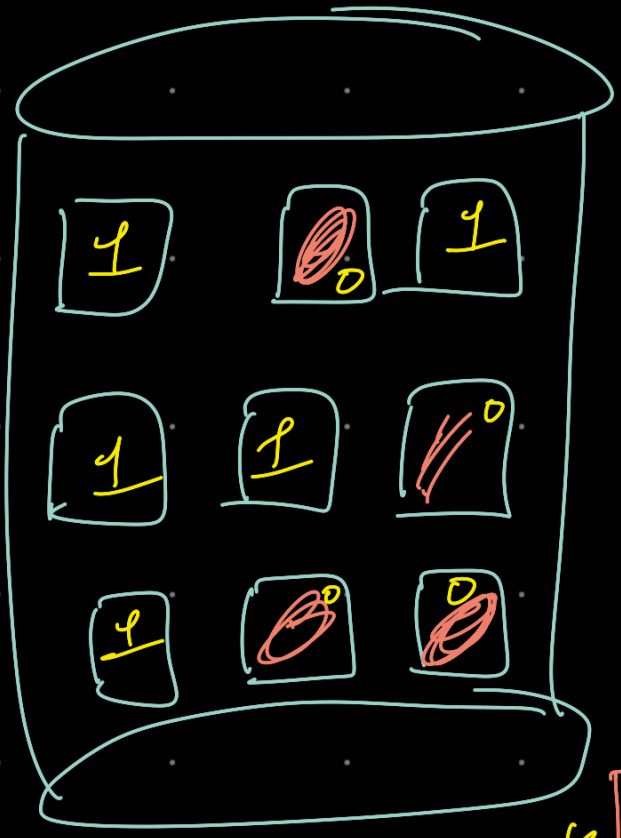
Linked list

Bit-map

linked list



Bit-map



$\Rightarrow$ no. of bits required

$\Rightarrow$ No. of physical sectors in HDD

$\Rightarrow$ also stored inside disk only