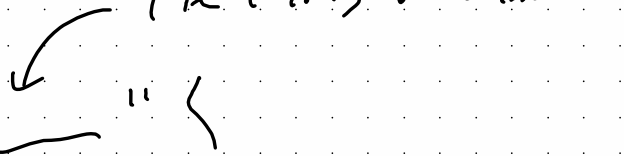
UPC code

Products	Name	~~Id~~	Description	Price
		[string]		

- Most expensive product?
- Are there missing descriptions
- List the products in Alphabetical order
- What is the price for a given UPC code

```
let records: Vec<Product> = ____

for (record in records)
{
    if (record.upc == "____") {
        return record.price
    }
}
```



SELECT price] ← what I want (outputs)

FROM products] ← where it comes from

WHERE upc = "~~"] ← what I want (records)

[ORDER BY] ←

$(10^3)^4 \approx 2^{40}$

1 Trillion ints (unique)

↳ Is an int part of that set

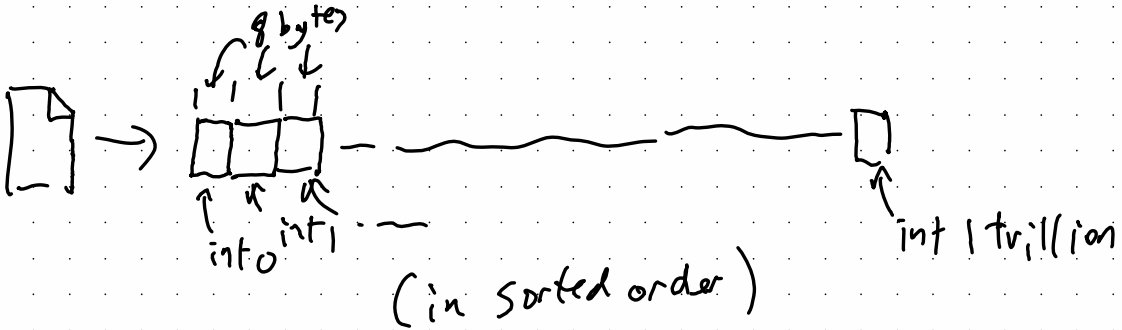
64bit

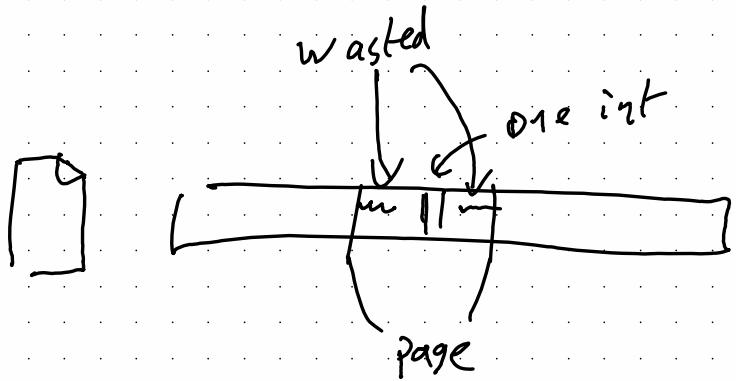
2^{40} ints = 2^{48} bytes $\frac{1}{4}$ Petabyte

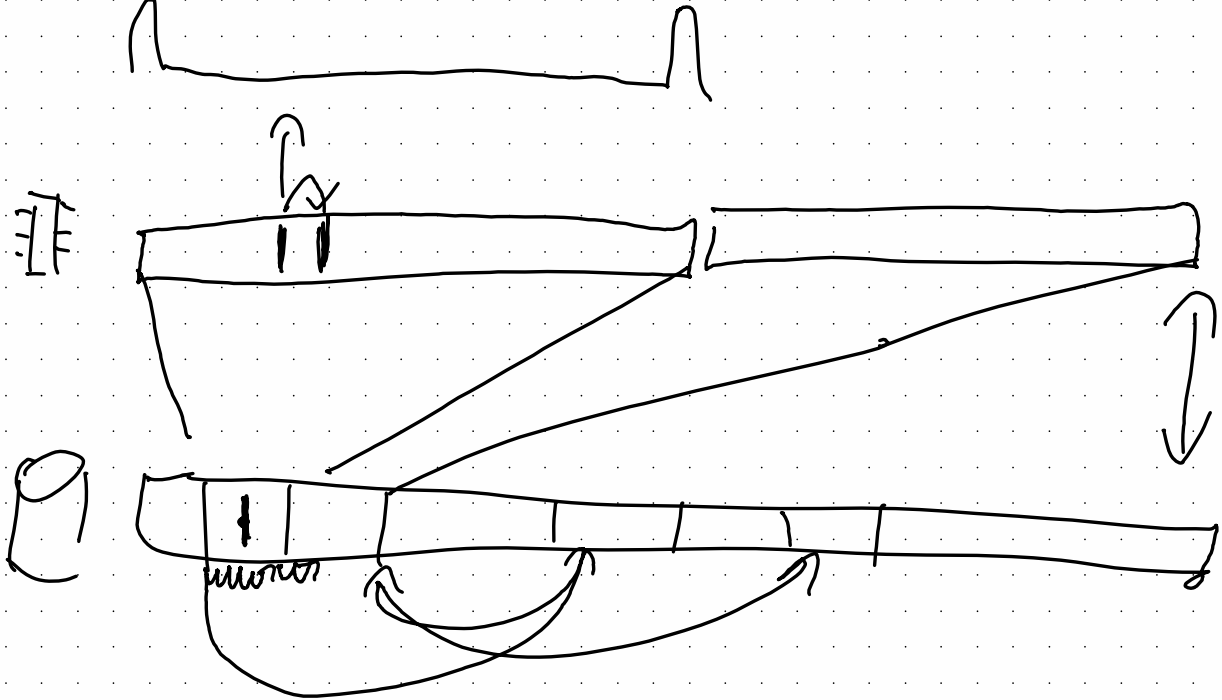
binary search over on-disk array

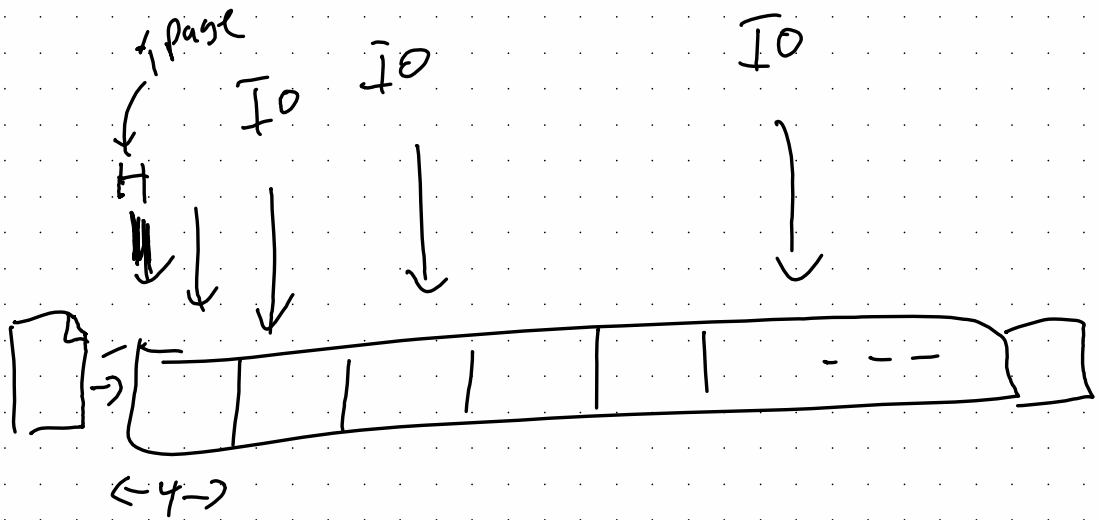
Assumptions

- we can sort the data



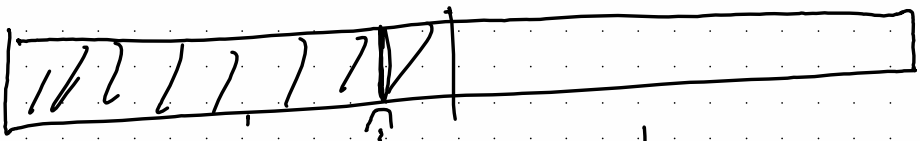

wasted
one int
page

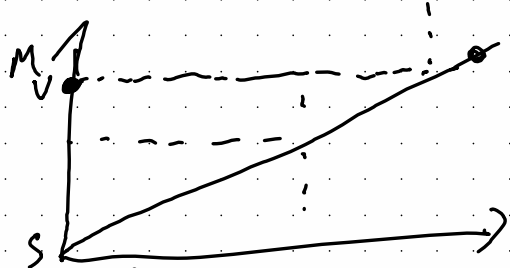




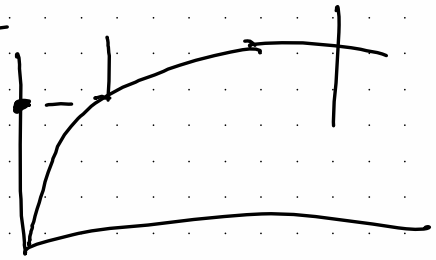
$$\#\text{ of IOs} = \text{Log}(N) - B$$

S > V C M F



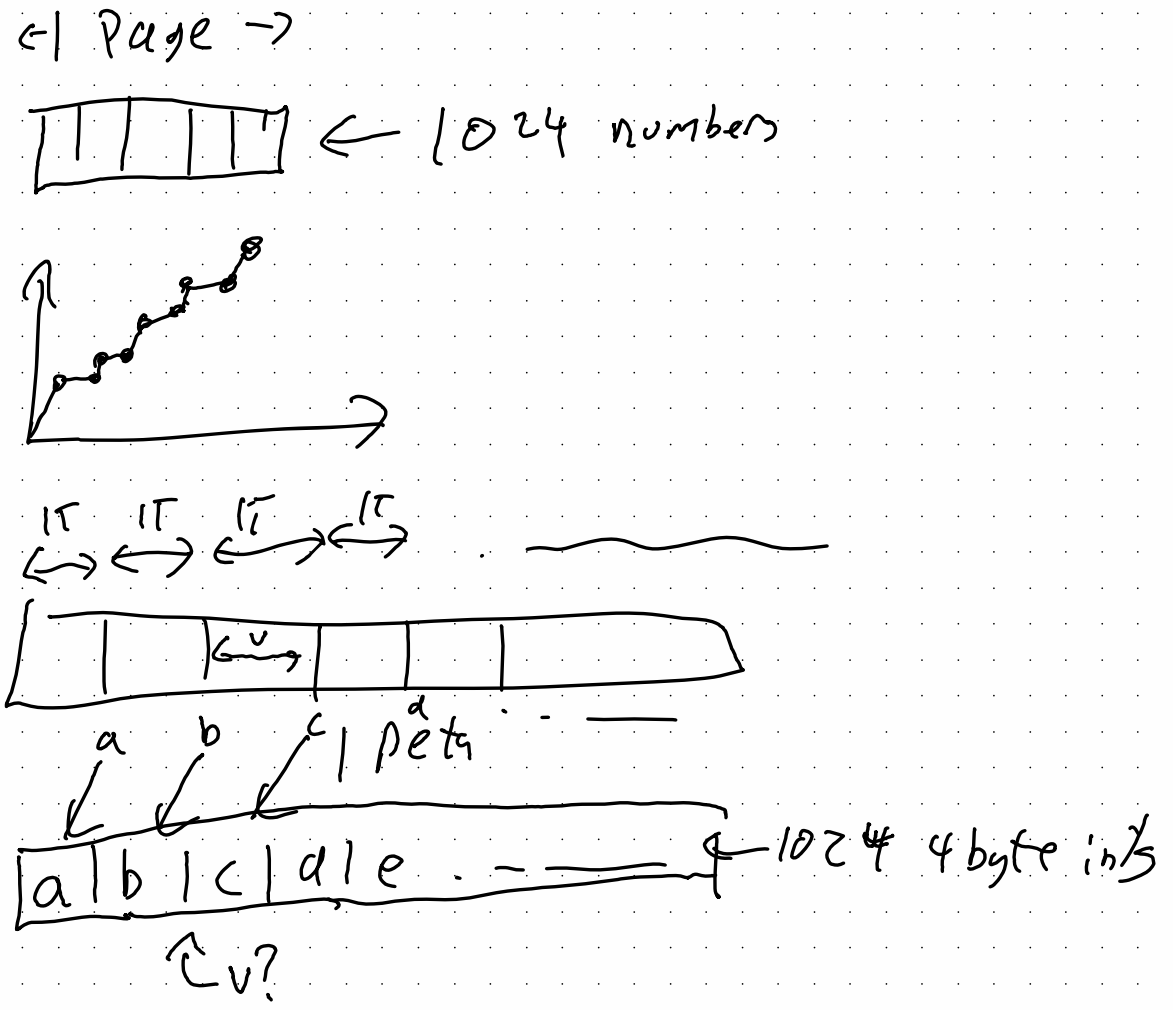


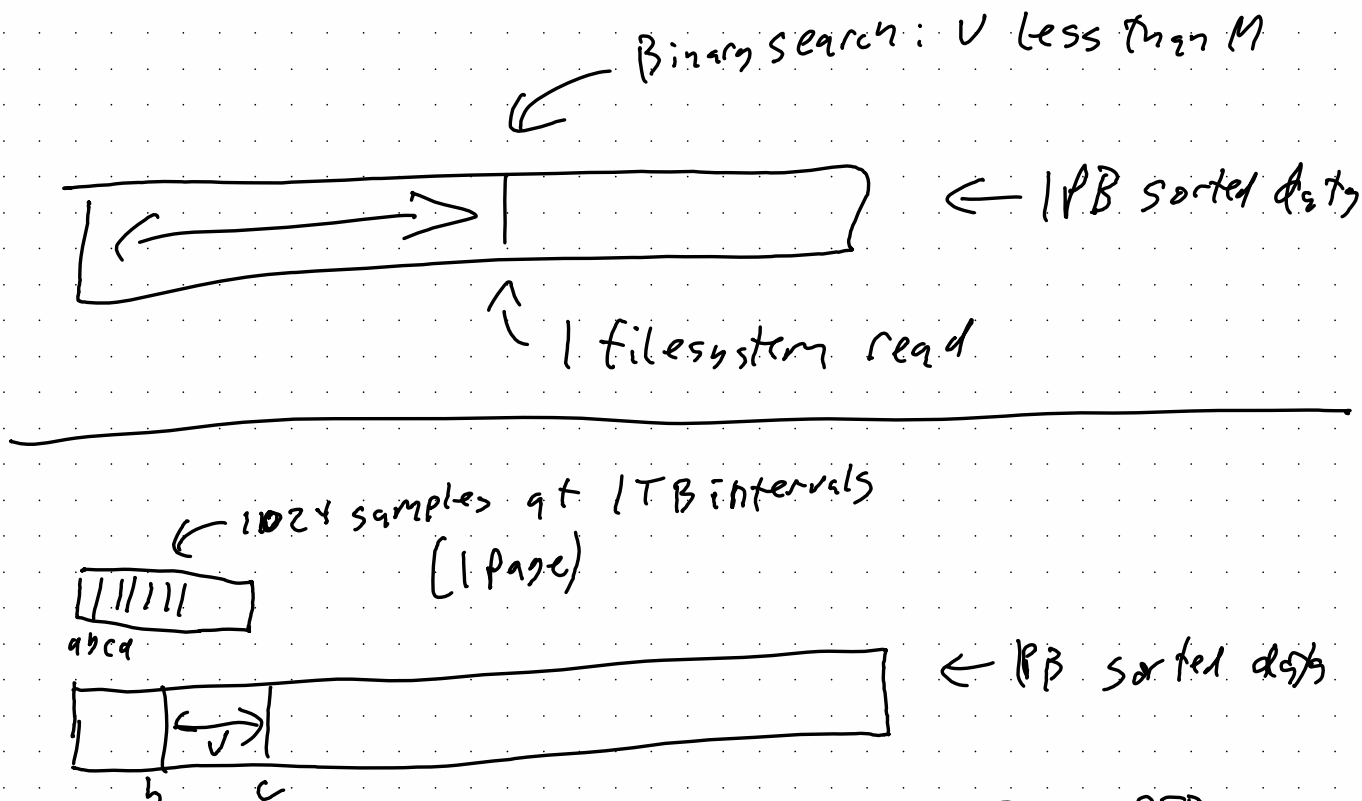
$y = mx + b$



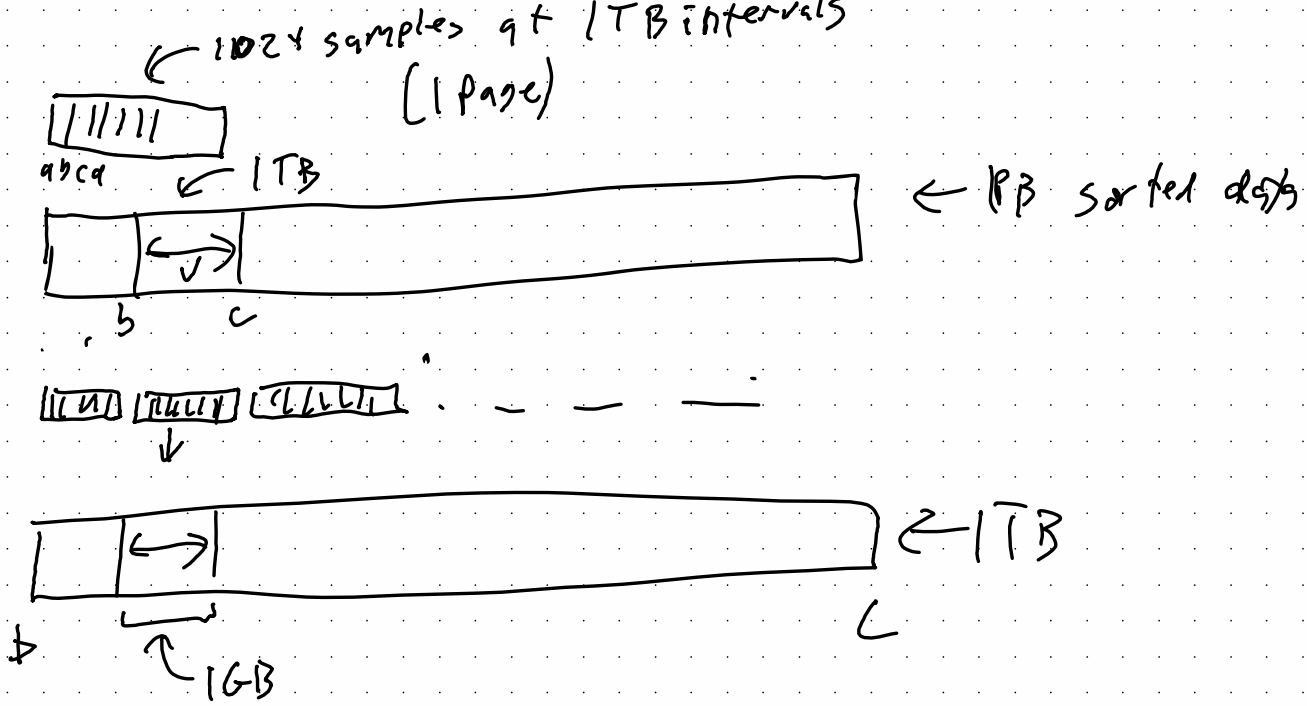
$y = a + bx + cx^2$

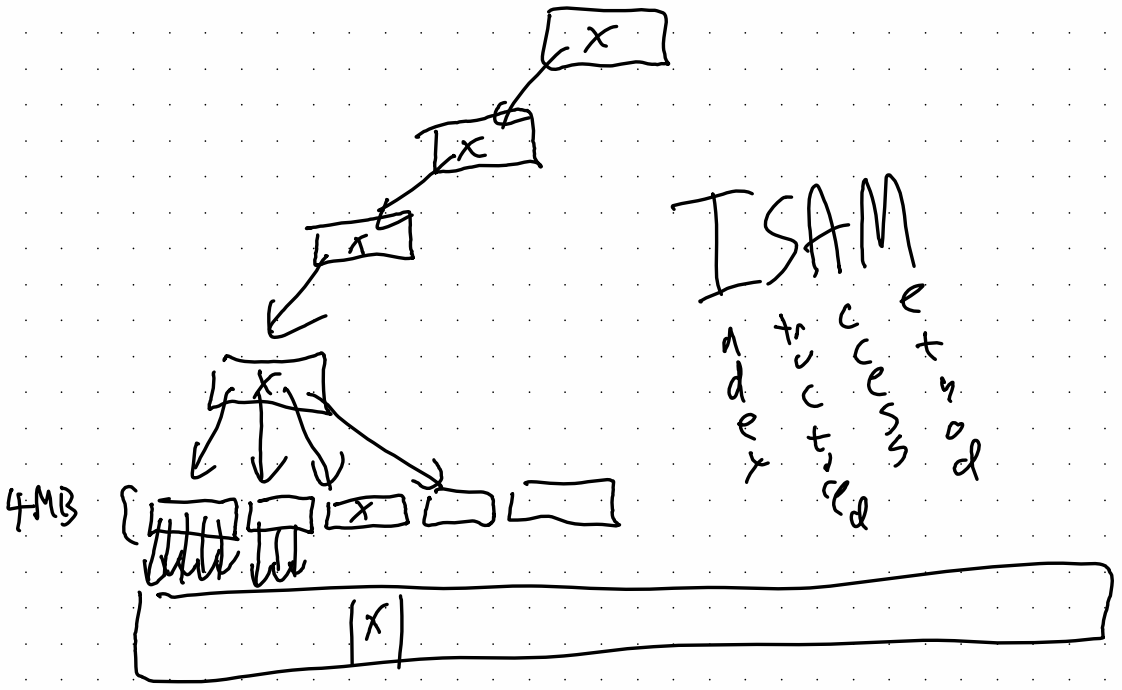
$y = \underbrace{f(x)}_{\text{CDF}}$


page
1024 numbers
IT IT IT IT
v
a b c d e
1024 4byte ints
v?



if v is between b c, then v must be in the 1TB – 2TB range


1024 samples at 1TB intervals
(1 page)
1TB
1PB sorted data
a
b
c
1TB
b
c
1GB



ISAM

index structured access method

1

10^8

10^{18}

10^{28} 1M pages

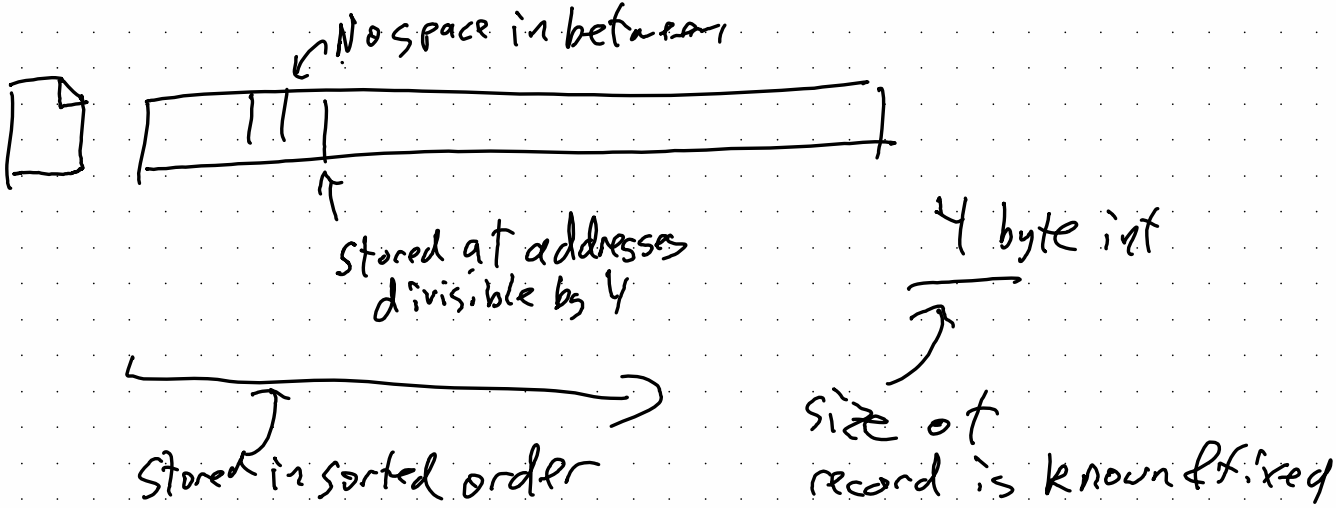
10^{38} 1k pages

1 PB

$O(\log_{1024}(N))$ vs $O(\log_2(N))$ for Bin search

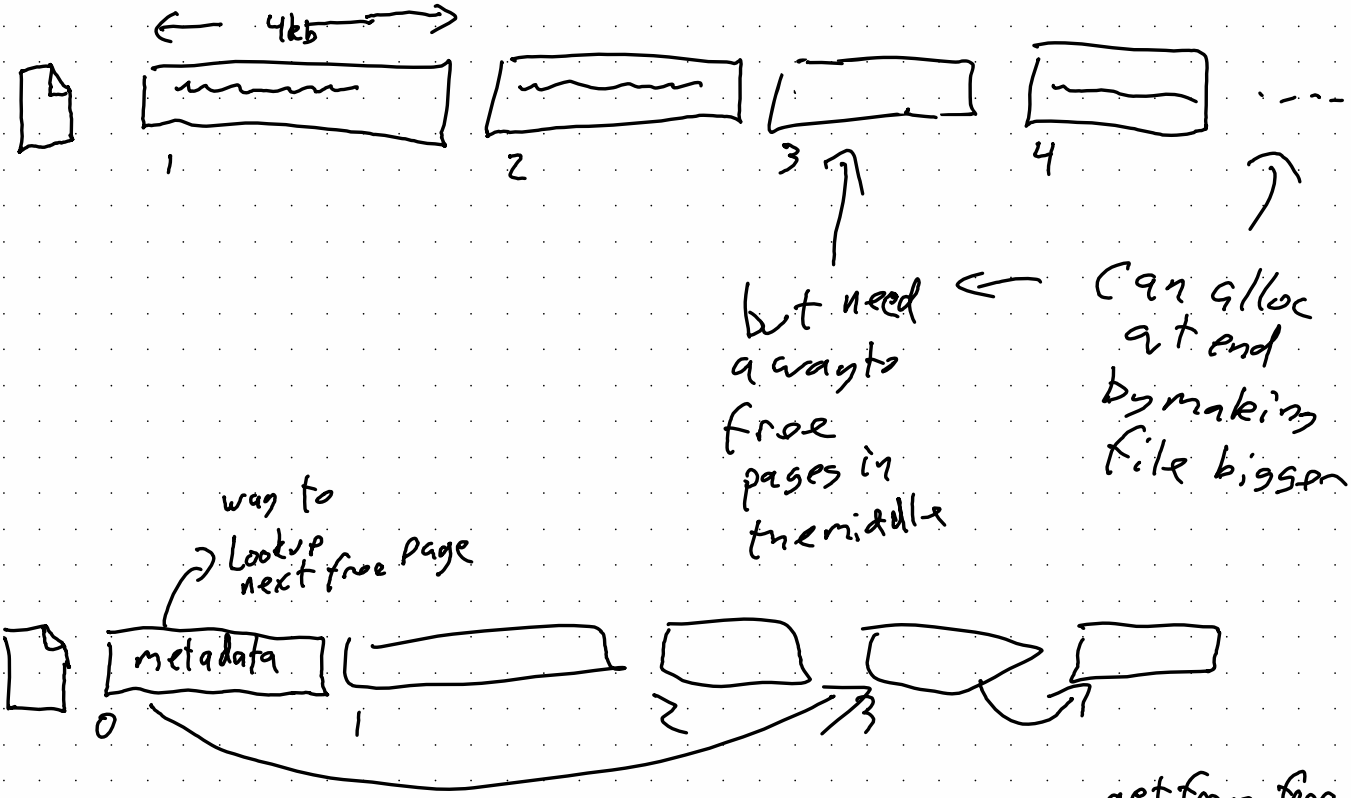
6 steps

2^{38}



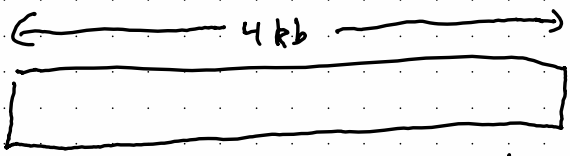
Problems

- No variable length records
- Expensive inserts/deletes



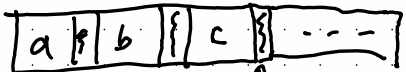
O(1) compute IO

- alloc page() → page id → get from free list / make file bigger
- free page(page id) → () → prepend to free list
- get page(page id) → start byte



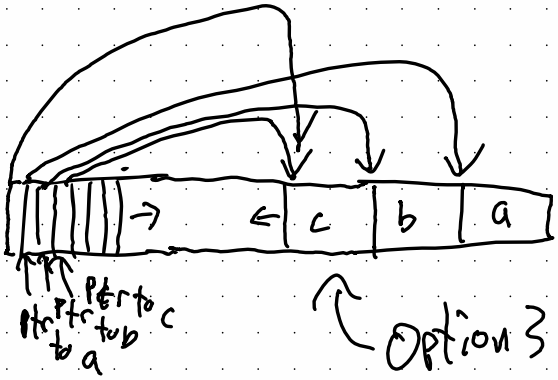
↖ Option 1: N-element array of records
where N · sizeof(record) < 4kb
↳ assumption: sizeof(record) is fixed and known

2 general assumptions
- sizeof(record) << 4kb
- each page is dedicated to one specific format



('\n' in CSV)

↖ Delimiter ('\0' for C strings)
Option 2: Delimited records

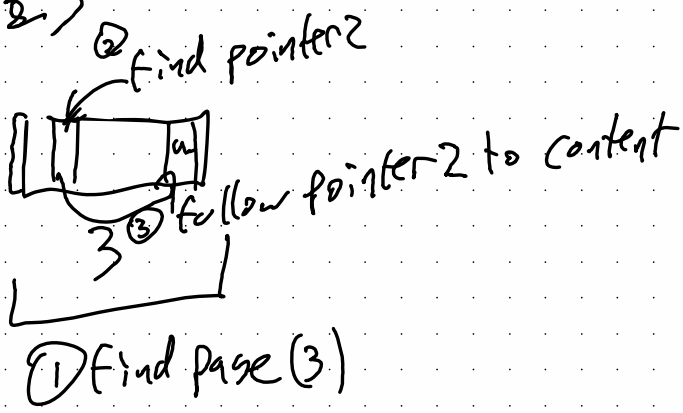


↖ Option 3: Array of pointers to first byte of variable-size record

Locating Records

- <Page #, Record #>
  - Array → Which pos in array?
  - Delimited → How many preceding delim?
  - Pointer → Which pointer to follow?
  - find the 4kb chunk in the file

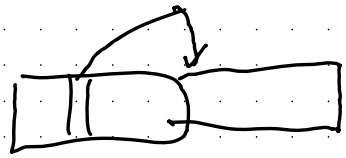
Get <3, 2>



Predictably sized fields → Option 1

Variable sized fields → option 2

Option 3 ← mix of both



variably sized component starts at byte 43