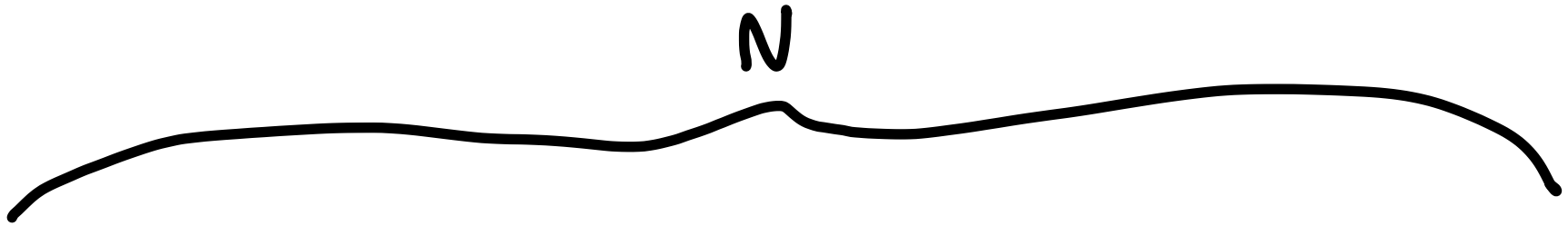
Big Data Algo

Lecture 9



$A = [2, 6, 10, 15, 22, 50]$

a

List — Looking for

```
def linearSearch(A, x)
    for a in A
        if x <= a
            return a
    return None
```

$x = 14$

0	1	2	3	4	5	6
1	3	7	10	12	16	18

l	r	m
0	6	3
4	6	5
4	5	4
5	5	

$x = 14$

```
def bs(A, x, l, r)          # l: 0, r: len(A)-1
    if l == r
        return A[l]
    m = (l+r)//2
    if x <= A[m]
        return bs(A, x, l, m)
    else
        return bs(A, x, m+1, r)
```

Linear Search $N \rightarrow 2N$

Runtime Double

$O(N)$

Binary Search $N \rightarrow 2N$

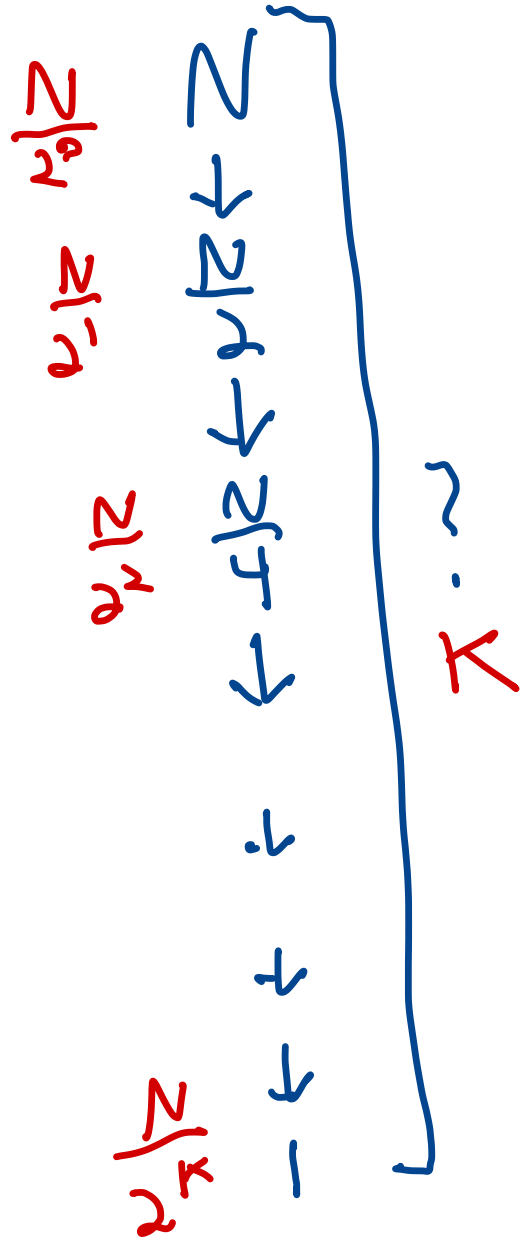
Runtime + Some constant

$O(\log_2 N)$

$$\frac{N}{2^K} = 1$$

$$N = 2^K$$

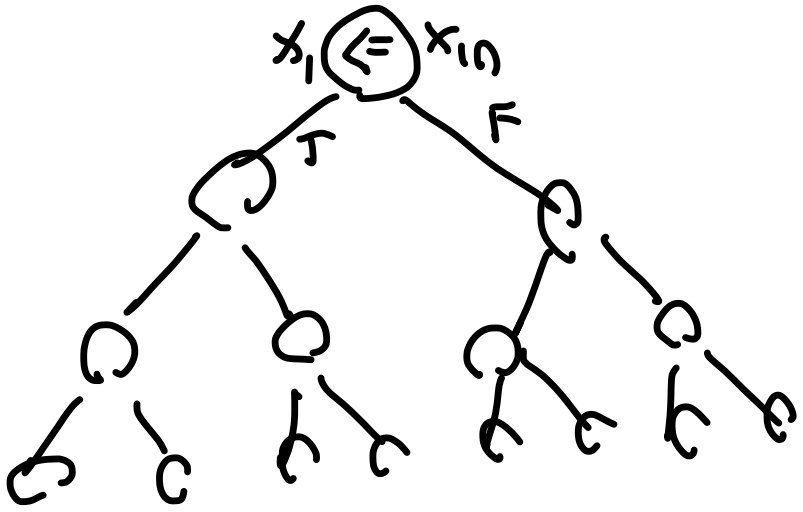
$$\log_2 N = K$$



Can You Do Better?

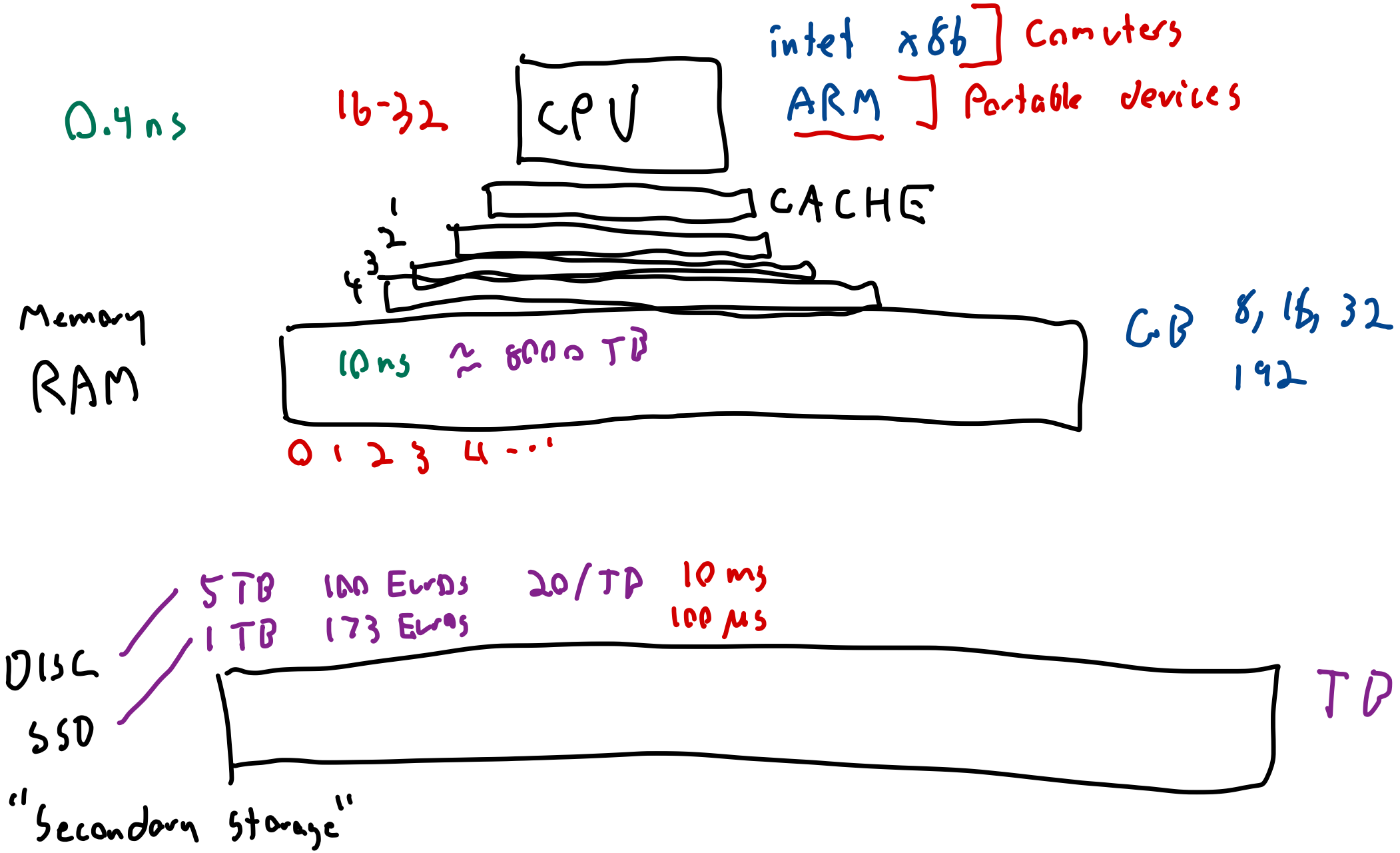
"Lower Bound" No

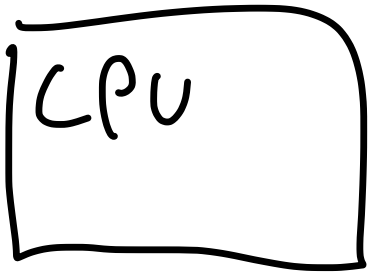
To search in N things requires at least $\log_2 N$ comparisons.



$2^C \geq N$

$C \geq \log N$


intel x86] Computers
ARM] Portable devices
0.4 ns
16-32
CPU
CACHE
1
2
3
Memory
RAM
10 ns ≈ 8000 TB
GB 8, 16, 32
192
0 1 2 3 4 ...
5 TB 100 Euros 20/TB 10 ms
1 TB 173 Euros 100 µs
DISC
SSD
TB
"Secondary storage"



Speed: 2.4 Ghz

$\frac{1}{2.4\text{Ghz}} = 0.4\text{ ns}$

Has ~ 16 - 32 "Registers"

Register: Place to store 1 Value (1 value 64 bits)

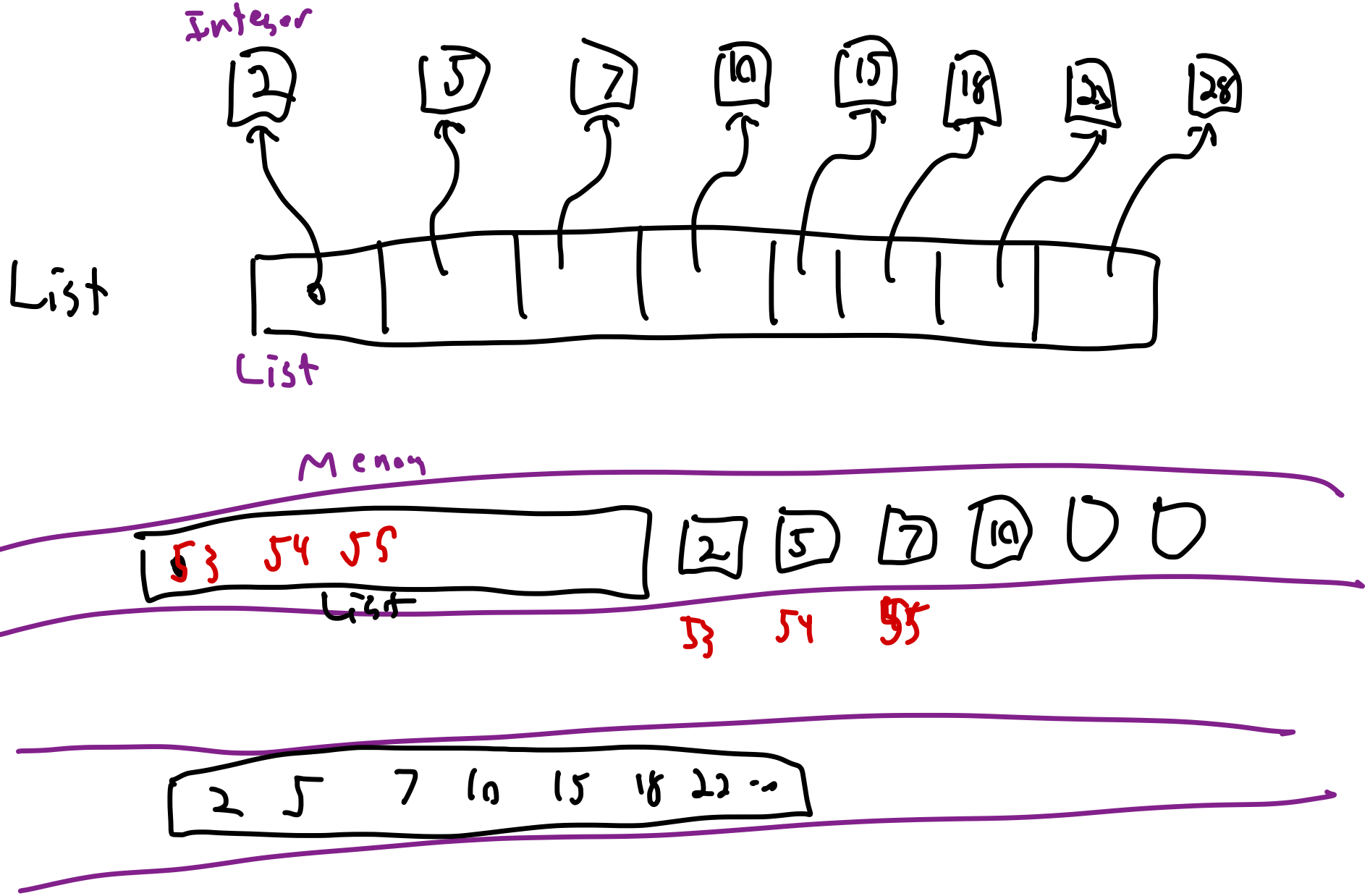
At each time step:

- Do math on the registers
- Read / Write a register from Memory

$ns = 10^{-9}\ s$

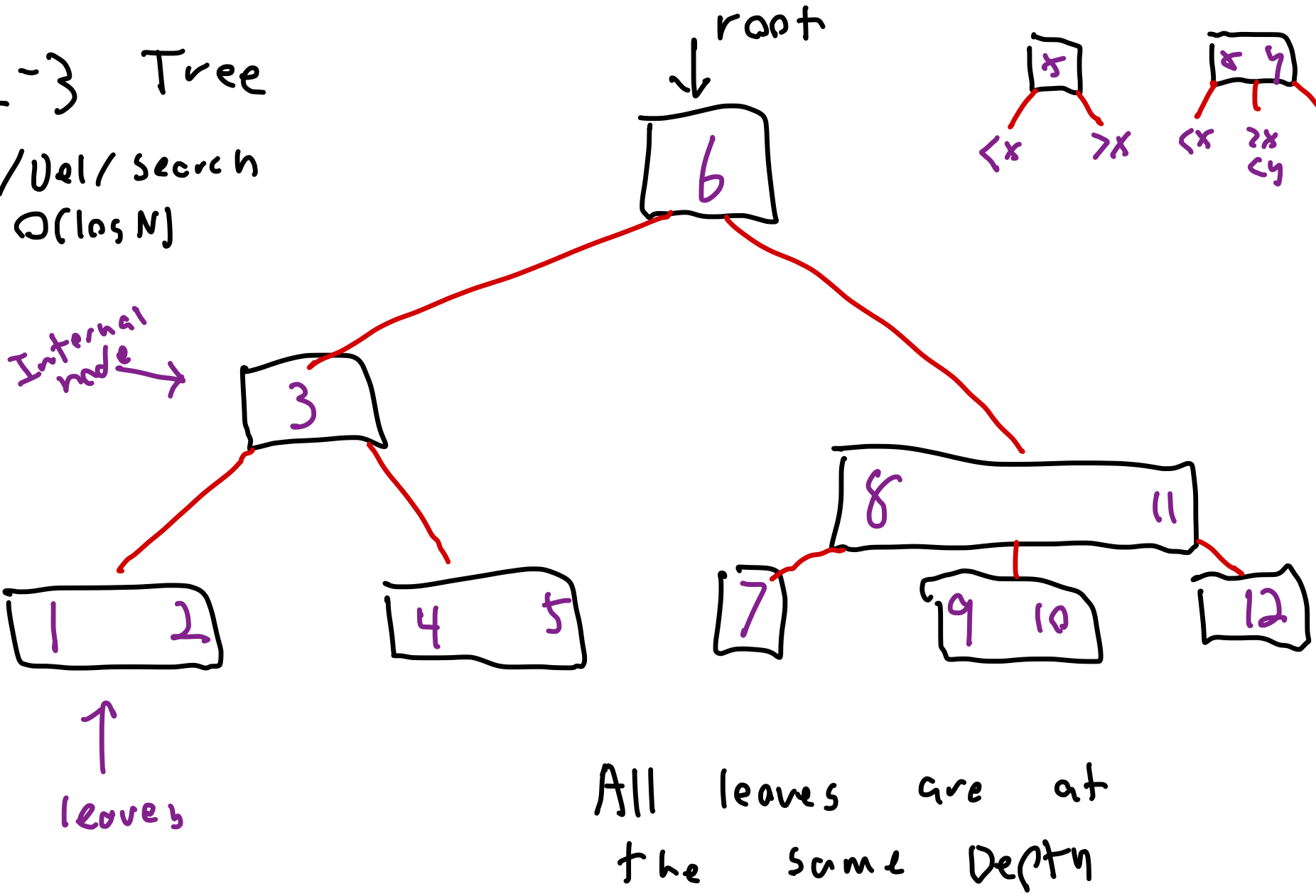
$\mu s = 10^{-6}\ s$

$ms = 10^{-3}\ s$


Integer
2
5
7
10
15
18
23
28
List
List
Memory
53
54
55
List
2
5
7
10
53
54
55
2
5
7
10
15
18
22

2-3 Tree

Ins/Del/Search
$O(\log N)$



All leaves are at the same Depth

1, 2, 3, 4, 5, ...

Class Tree

Class Node

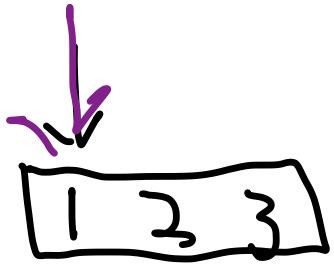
[Data
[children

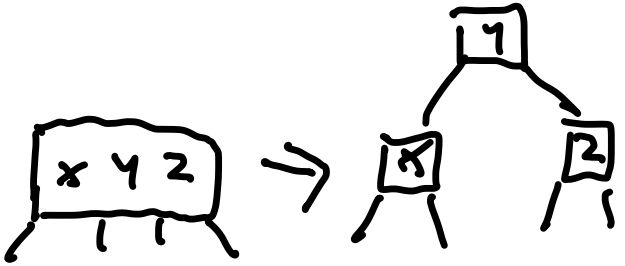
root

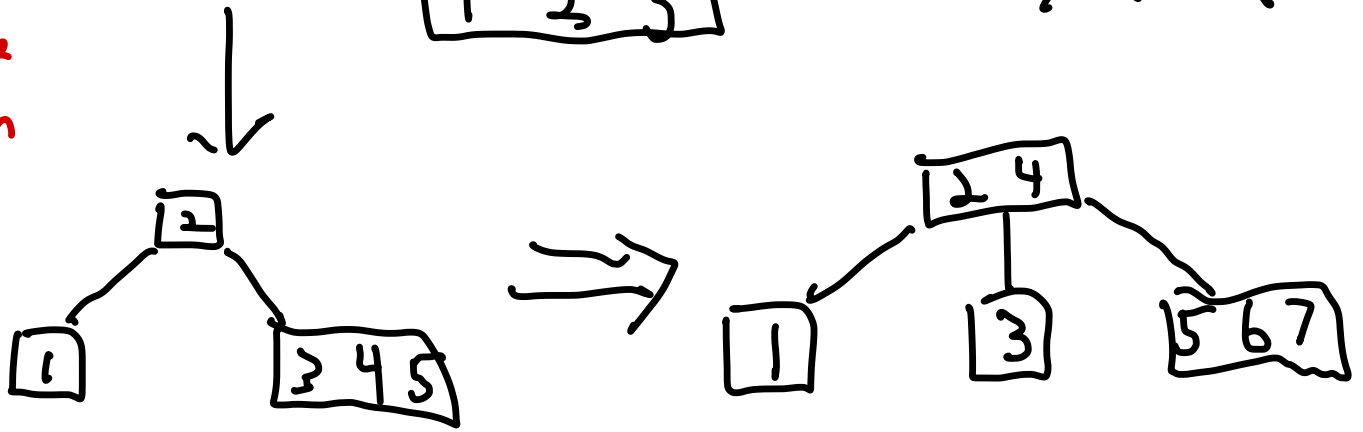
Insert

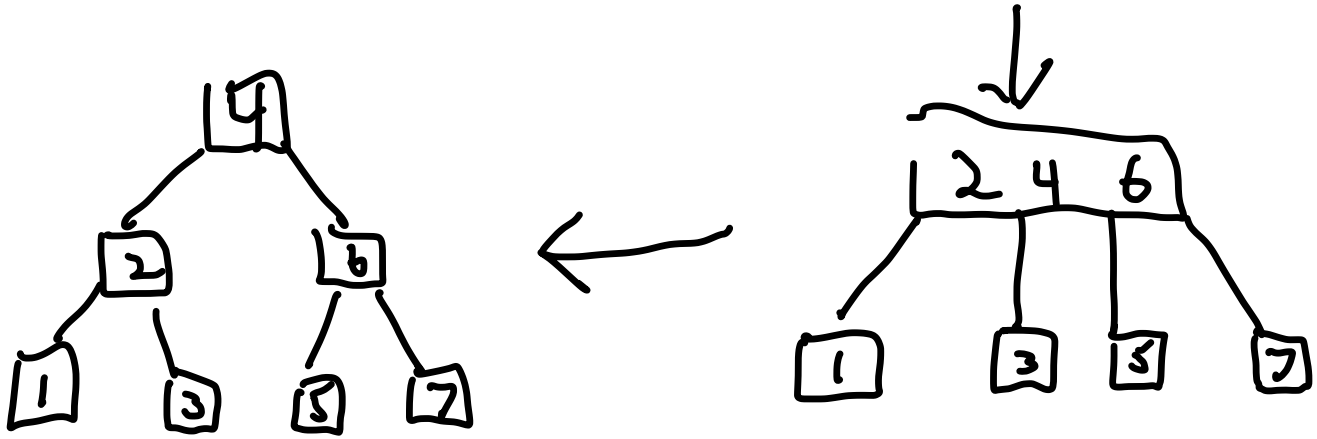
Delete

Search









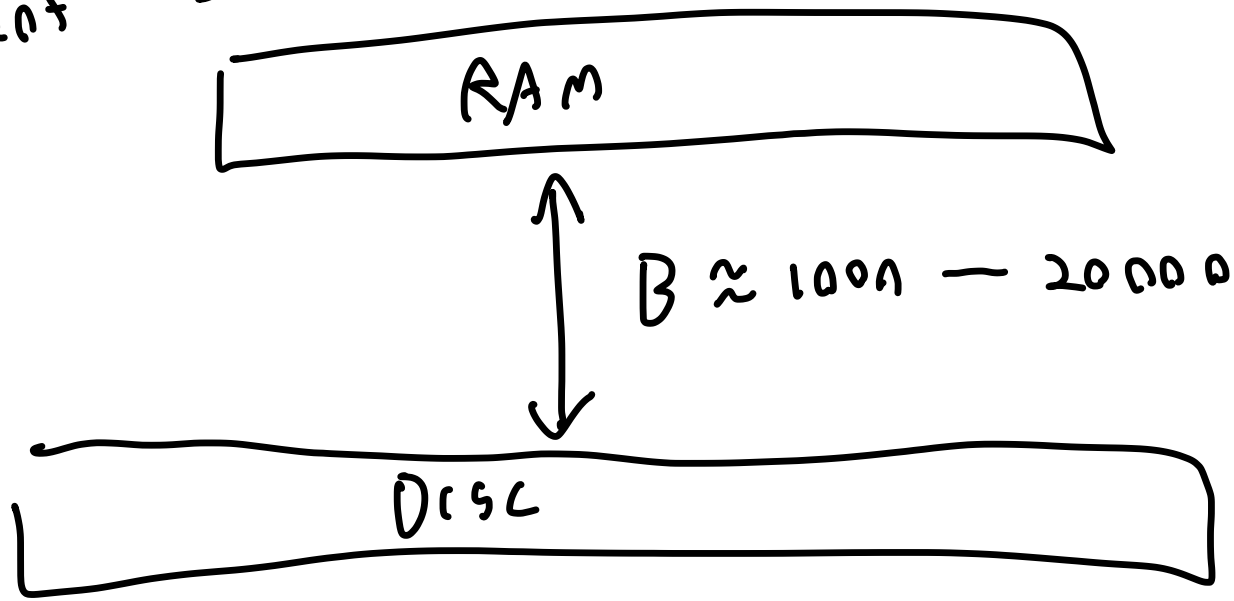
B-Tree

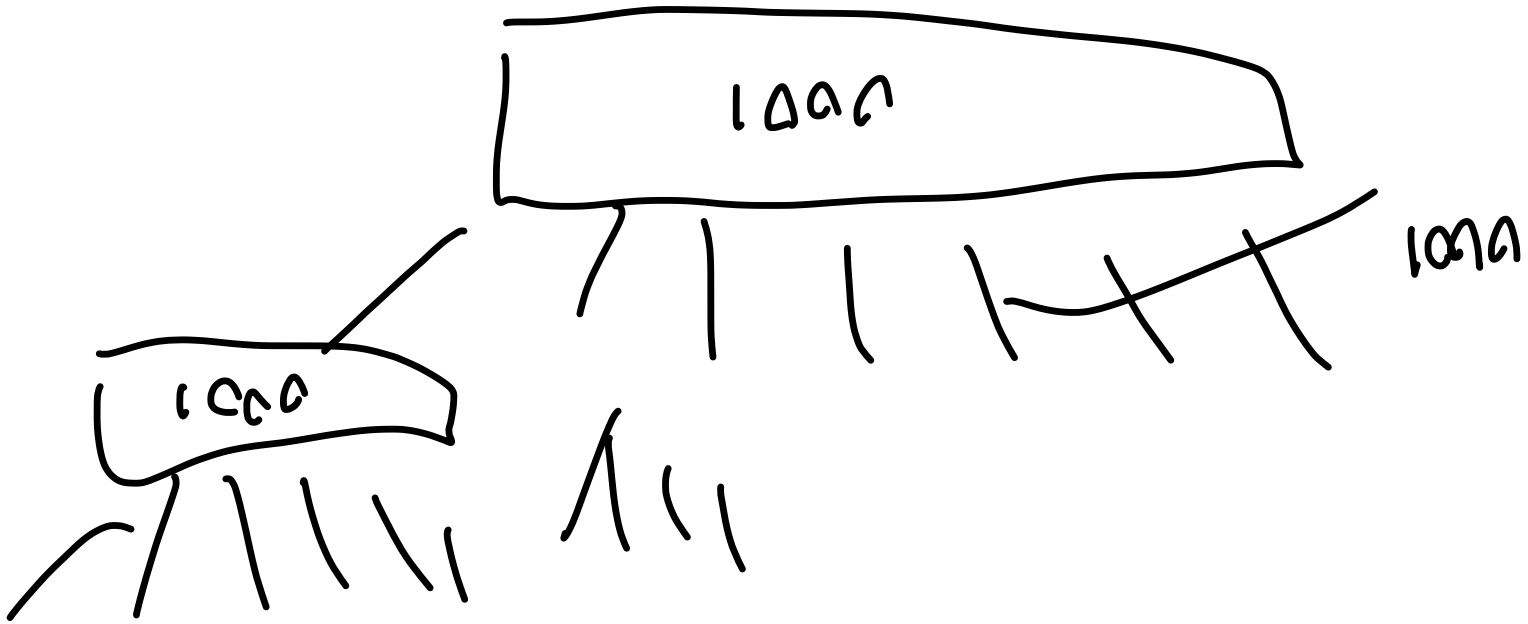
= 2-3 Tree

Except $B_1 - B$

1,000,000

$\log_2 1{,}000{,}000 \approx 10$




1000
1000
1000