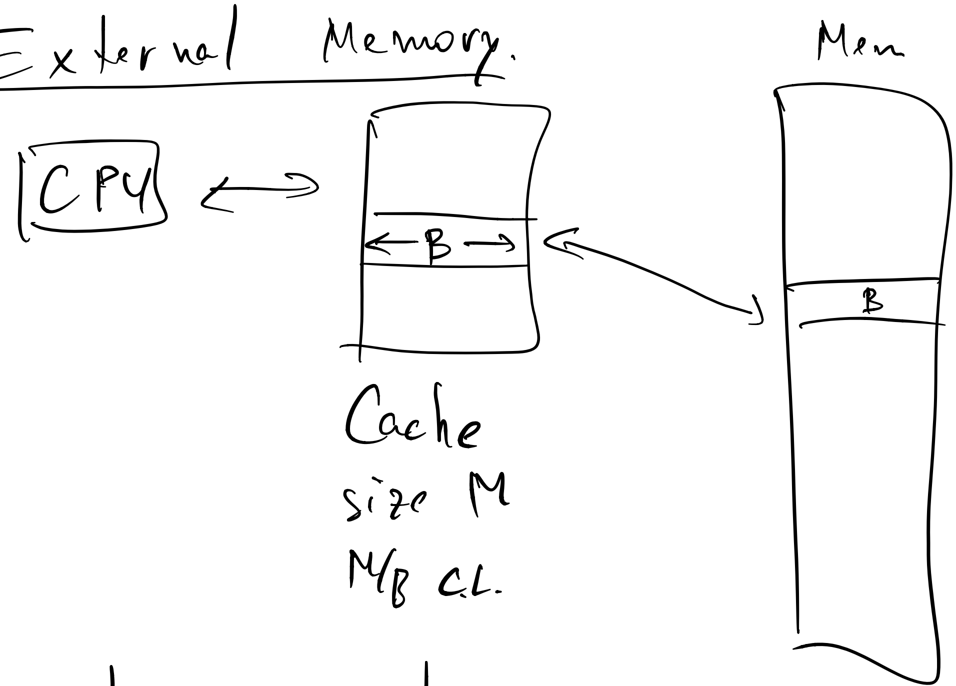
AA Lecture 26 4/15/21

I/O External Memory.



Cache
size M
M/B c.l.

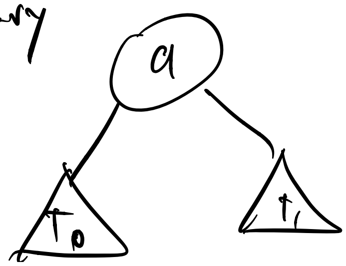
Problems binary search
searching in ordered domain.
B.S.T.

Naive BS: $\Theta\left(1 + \lg \frac{n}{B}\right) = \Theta(\lg n)$

$B \ll \sqrt{n}$

Cache version: B-trees.

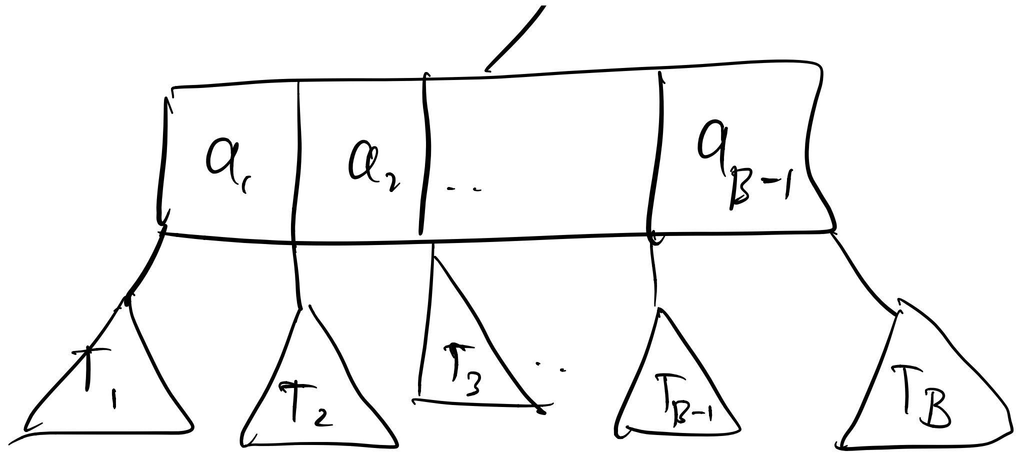
2-ary



$\forall\ x \in T_0 :\quad x \leq a$

$\forall\ x \in T_1 :\quad a \leq x$

B-ary



Props: $\forall x \in T_1 \le a_1 \le \forall x \in T_2 \le a_2 \le \dots$

$\dots \le a_{B-1} \le \forall x \in T_B$.

Search Algo: - natural search in the tree

- if searching x:

if $x = a_1 \dots a_{B-1}$, output

else find i s.t.

$a_i \le x \le a_{i+1}$

$(a_0 \triangleq -\infty$

$a_B \triangleq +\infty)$.

- recurse in T_{i+1}.

#CLM: proportional to depth of tree

$\hookrightarrow O(\lg_B n) = O\left(\frac{\lg n}{\lg B}\right).$

Eg. $B = 2^{10}$

$n = 2^{20}.$

naive: $\lg n/B = 20 - 10 = 10.$

B-tree: $\frac{\lg n}{\lg B} = \frac{20}{10} = 2.$

Remarks: CPU time for B-tree

naively: $O(B \cdot \lg_B n)$

$= O\left(\lg n \cdot \frac{B}{\lg B}\right).$

use binary search in each node

to find $i \in \{0, 1, \ldots, B-1\}$ s.t.

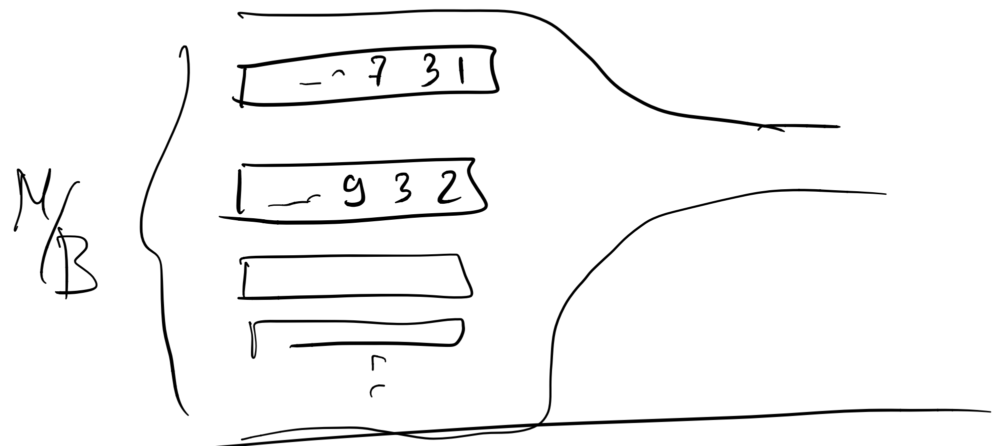
$a_i \leq x \leq a_{i+1}.$

CPU time: $O(\lg B \cdot \lg_B n)$

$= O(\lg n).$

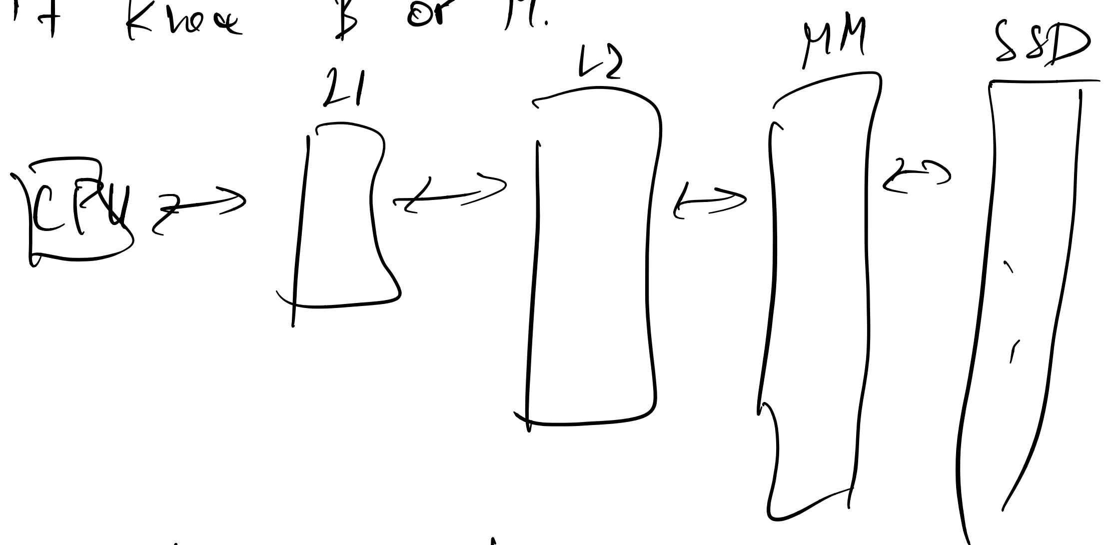
Fact: Sorting: $O\left(\frac{n}{B} \cdot \lg_{M/B} \frac{n}{B}\right).$

idea: use M/B-way merge sort.



Cache-Oblivious Model

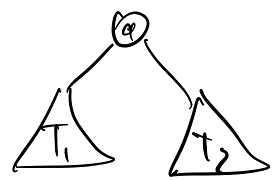
Don't know B or M.



Problem: binary search

van emde Boas layout: $O(\lg_B n)$ CLM.

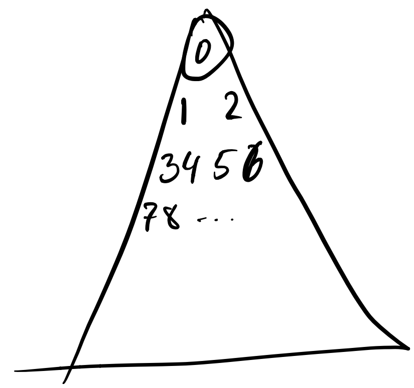
binary tree:



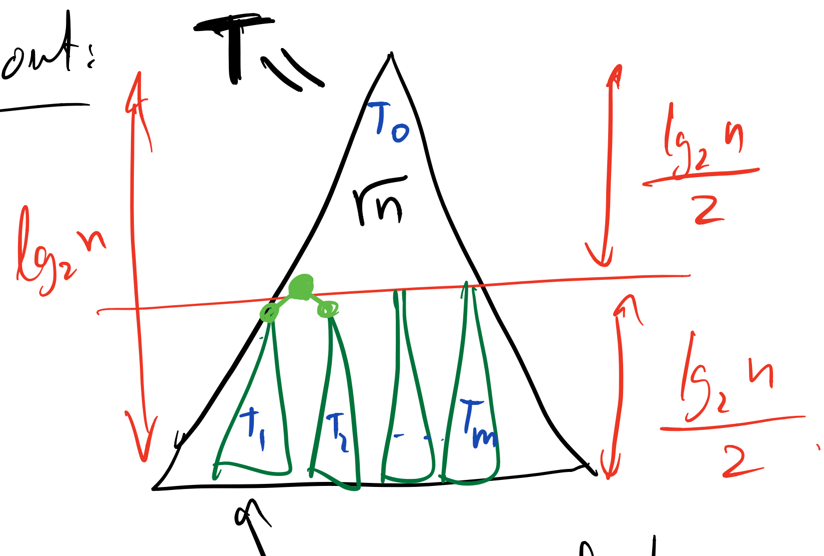
$T_1 \le a \le T_2$.

standard layout of bal. bin. tree: <u>heap</u>.

array $A[0..n-1]$, element i has children @ $2i+1$, $2i+2$.



<u>veB layout:</u>



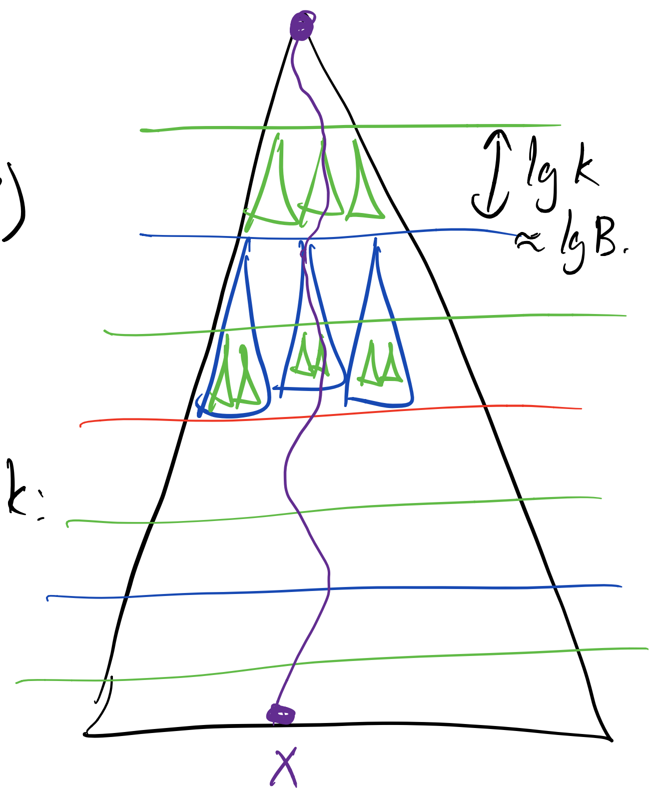
about $\sqrt{n}$ of trees of size $\sim\sqrt{n}$.

$veB(T) =$ • if $n \le 16$, store explicitly.

- otherwise:

$$= [\mathrm{veB}(T_0), \mathrm{veB}(T_1), \dots \mathrm{veB}(T_m)]$$

#CLM for veB?



fix level ℓ. (of veB)

tree size: $n^{(\frac{1}{2})^\ell} = k$.

CLM per tree of size k:

$O(1 + \frac{k}{B})$.

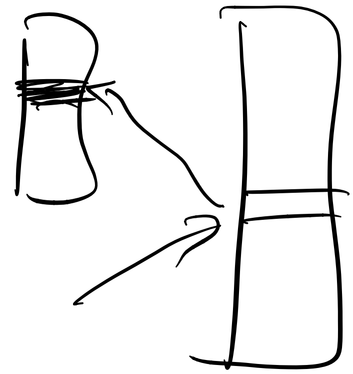
fix $k \simeq B$. $\left(\text{fix } \ell \text{ s.t.: } \sqrt{B} \leq \underset{k}{\underset{\|}{n^{(\frac{1}{2})^\ell}}} \leq B\right)$

total $\#\mathrm{CLM} = O\left(\frac{\lg n}{\lg B}\right) = O(\lg_B n)$.

Cache Line Transfers.

Cache Mem

Alg Opt: optimally, even knowing "future"



Alg: FIFO:

LRU:

Theorem: algorithm A with opt. paging uses T time (CLM) on cache of size M. Then run A with FIFO/LRU on cache of size 2M will have time $\le 2T$.

competitive ratio of online algo.

with resource augmentation.