5112

10-24

Models

Why algorithms are [not] useful?

RAM Model

An array of memory 

Have some registers 

Operations: Load a word from memory into a register
Store " " from a register into memory
Arithmetic: +, −, ×, ÷, maybe other things.

} each cost $\Theta(1)$

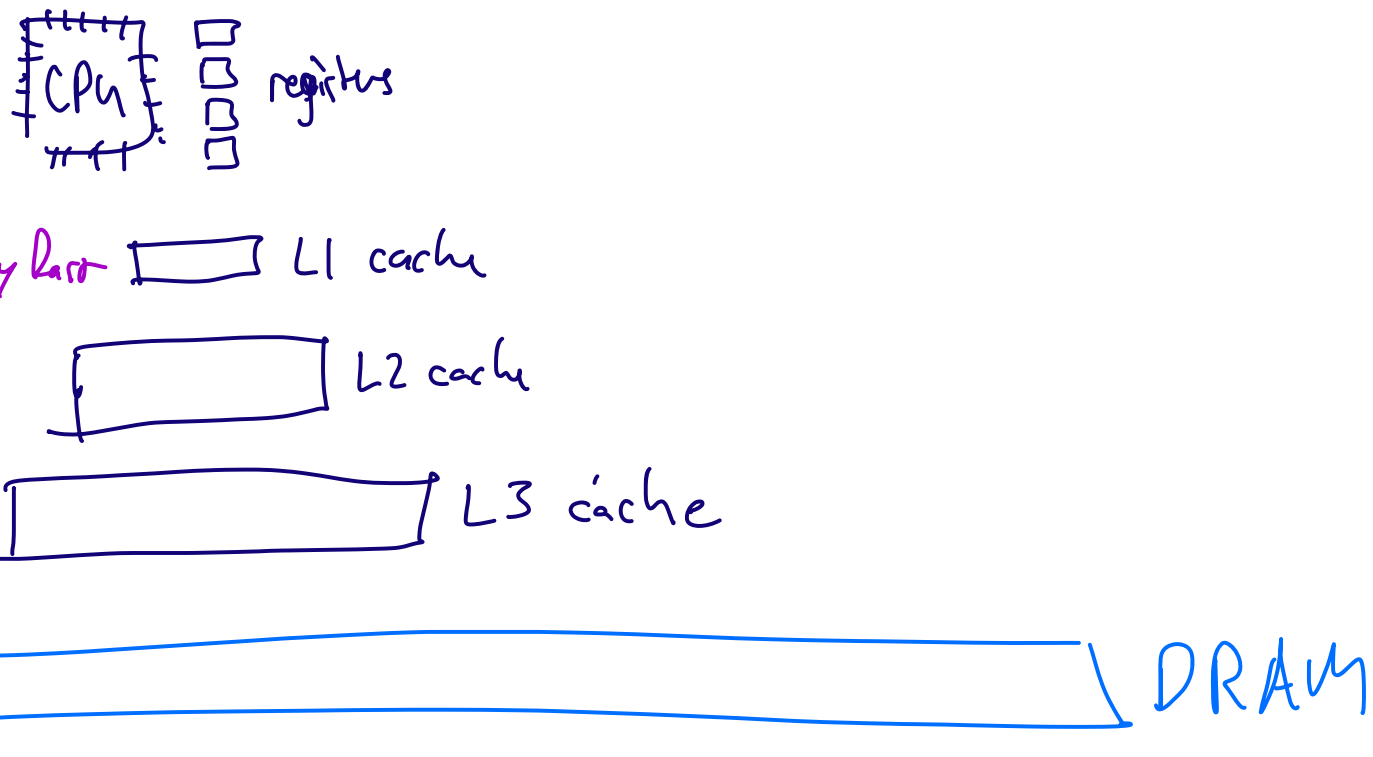
What make a good model?

A good model predict performance.

1. Accurate
2. Useable

} in tension with each other

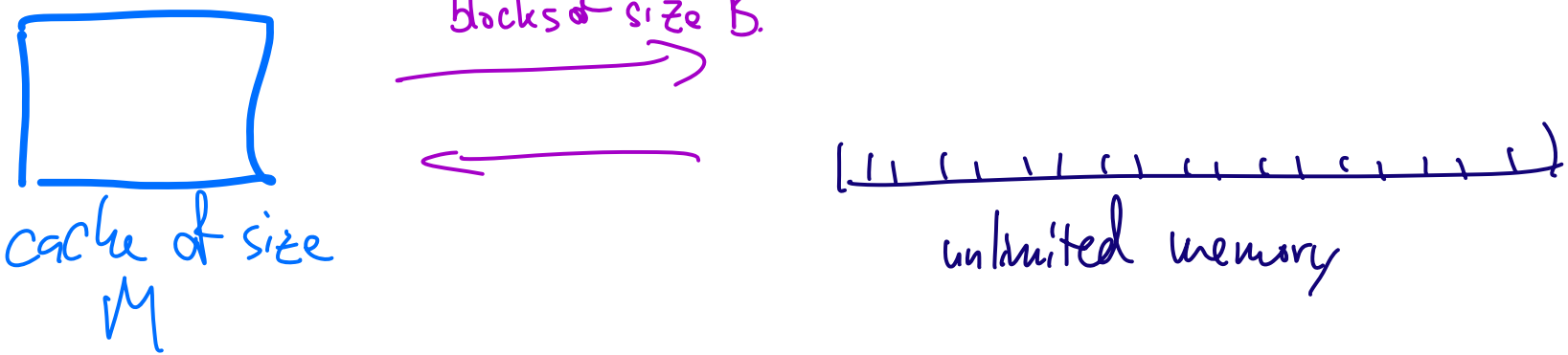
RAM Model and Memory



Cell Probe Model

Same as the RAM model, but all operations on registers are free.

External Memory Model



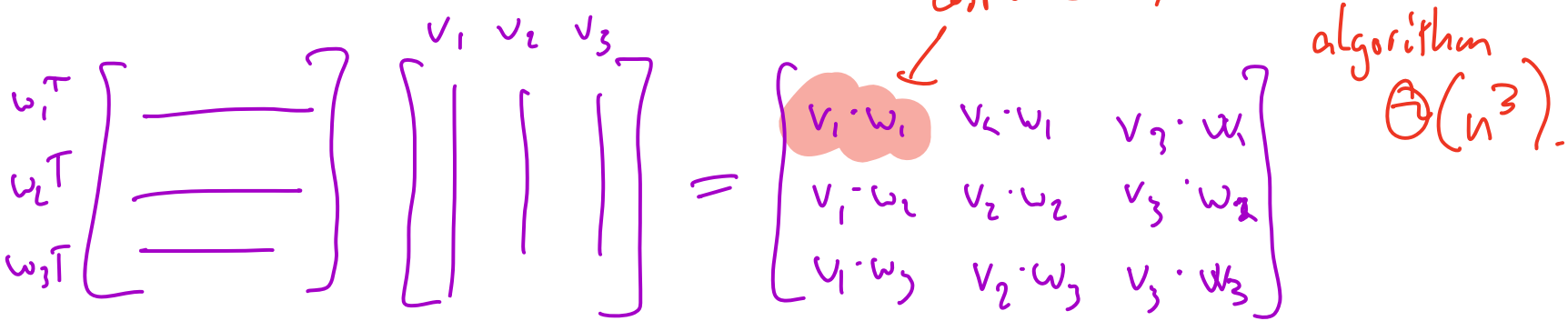
Everything in cache is free.

Cost is the # of transfers.

Asymptotic Analysis.

Accurate? Useful?

Is it accurate? It say something about performance of "large enough" inputs.



Models of Data

Example: sorting

"Sorting costs $\Theta(n \log n)$"

Implicitly in the comparison model.

But what if you are sorting integers?

Give me n integers $\leq U$.

A 

length U.

For each key k, put it into the k^{th} entry ($A[k]$)

Then read through the array.

Cost : $O(n + U)$

Counting Sort

Radix Sort

3 decimal integers

Start with least significant digit.

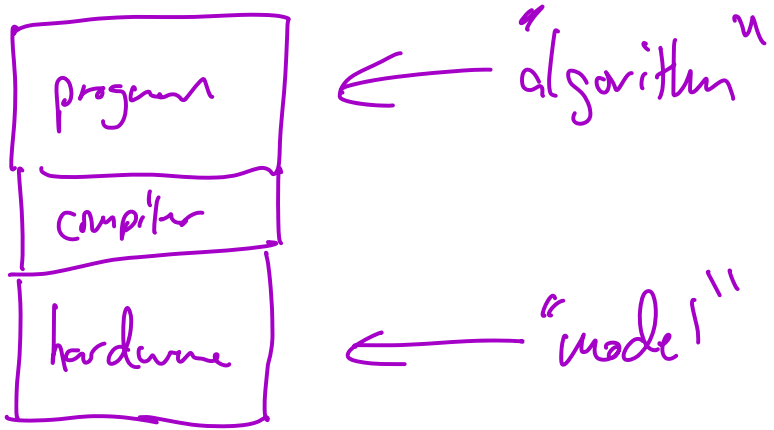
	109	351	464	521	700	718
	700	351	521	464	718	109
sort by 2nd digit →	700	109	718	521	351	464
	109	351	464	521	700	718

sort stably by LSD
" " " middle digit
" " " MSD

Suppose the radix is b. $n+b$ per round and u/b rounds

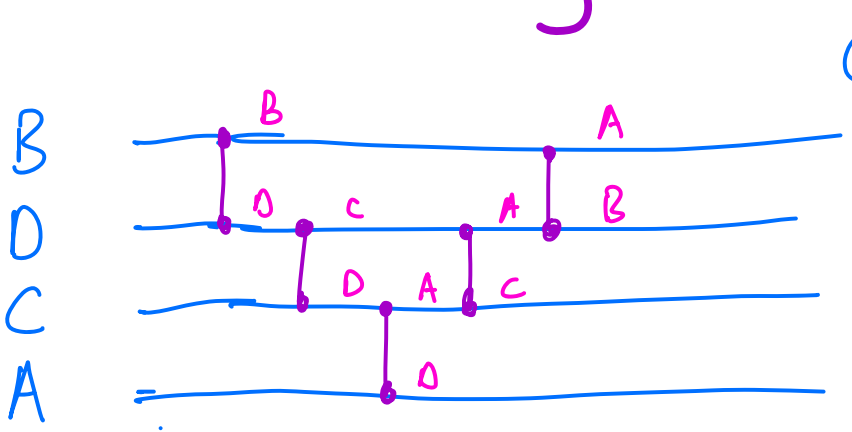
Cost: $O(u(n/b + u))$

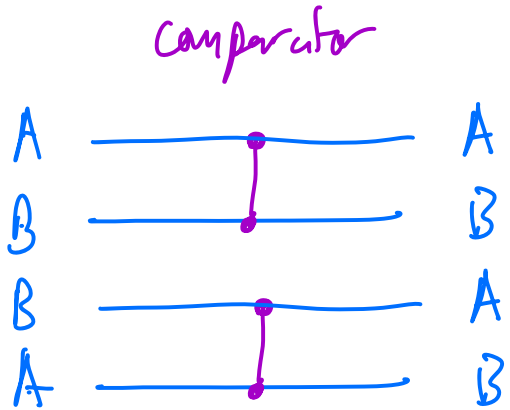
"Usual Algorithms Universe"

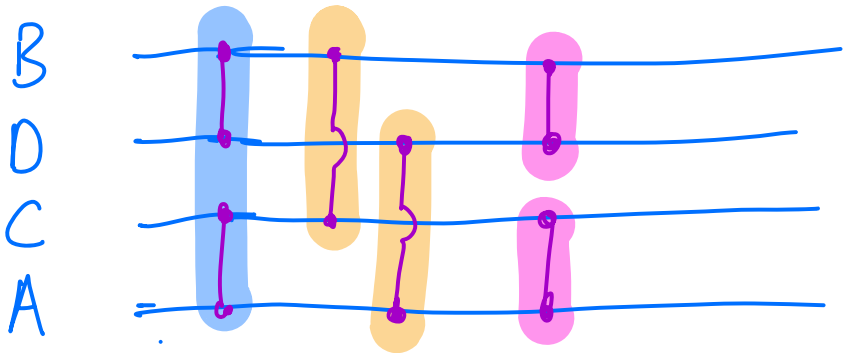


Can also look at hardware algorithms

Sorting Networks







output in sorted order