

CS2100

---

Recap day


---

---

---

---

---



# Final announcements

- HW due
- Review session Monday
  - bring questions!
- Final: Wed. at 8am, here
- Keep an eye on blackboard/git for more grading
- Today: have to sub 1-2pm in my office at 2pm
- Request: instructor evals!  
(You'll have time at end)
- Tuesday: In, but likely down in linux classroom.)

# Data Structures we've seen:

- Stacks + queues
- simple (singly linked) list
- Vectors
- Lists
- Trees:
  - general, BST, AVLs, heaps, Huffman

definitely

- Hashing
- Graphs
- Treaps

definitely

Also:

- C++ - tons! (low-level)
- Sorting / searching

# Trade-offs:

## Simple + limited:

- stacks
- queues }  $O(1)$
- even priority queues  
(heaps)
- hashing

## Why use?

Fast!

Overhead savings.

"Full-featured":

- Vectors
- Lists
- Trees

Trade-offs are key!

Consider:

- your data
- ~~a~~ how you'll use it

## Practical vs theoretical:

Some have poor theoretical guarantees, but are amazing in practice.

- hashing

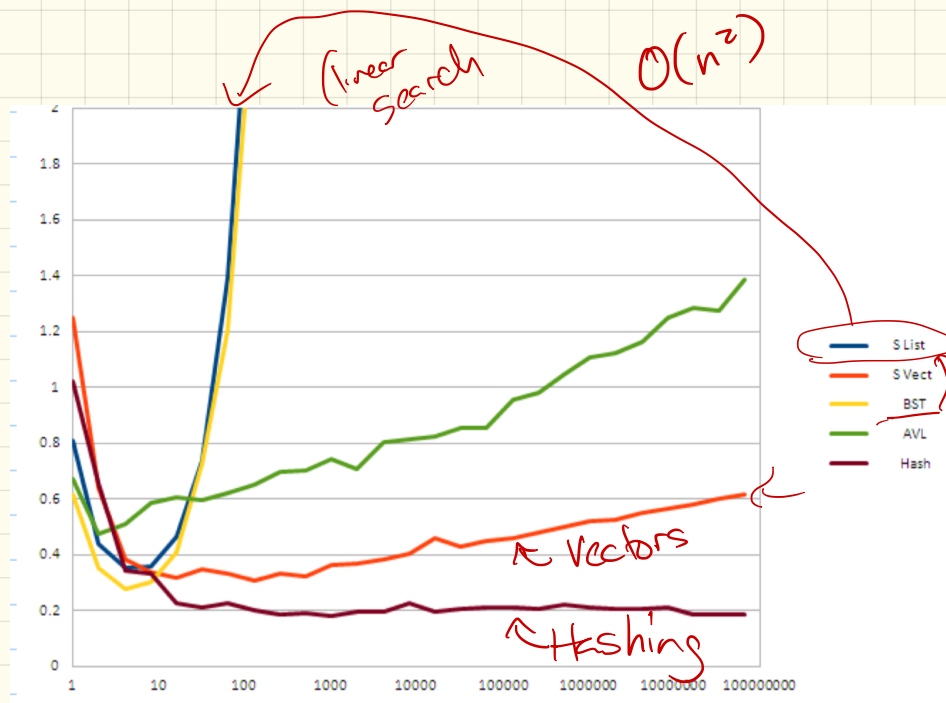
- quicksort

- even inserting in a vector  
(well-push-back)

Some data :

Insertions done in-order:

1, 2, 3, ..., n

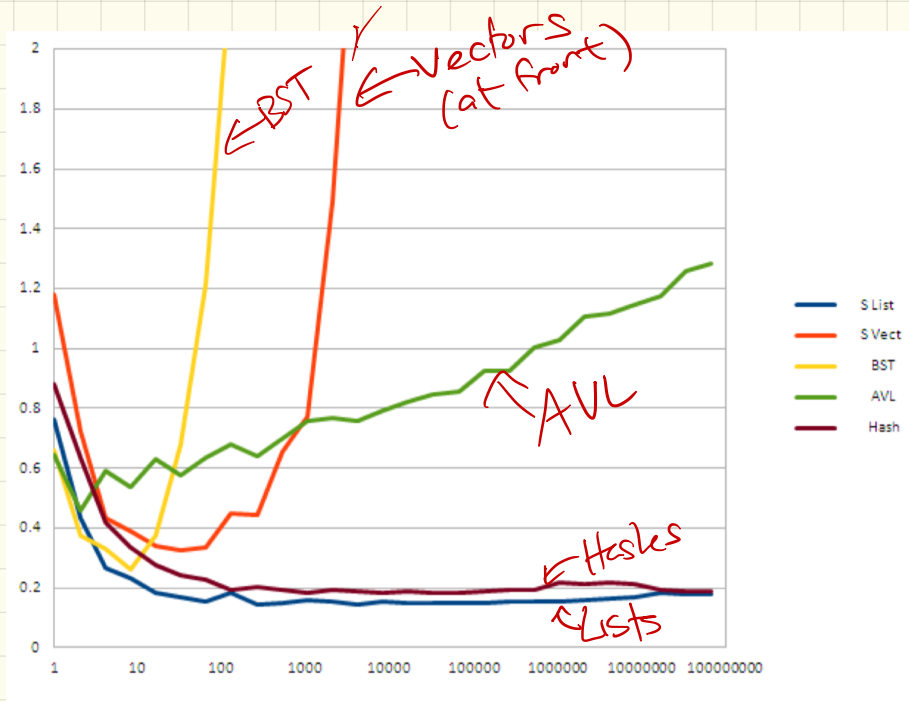


↑  
time

n grows →

# Reverse order inserts

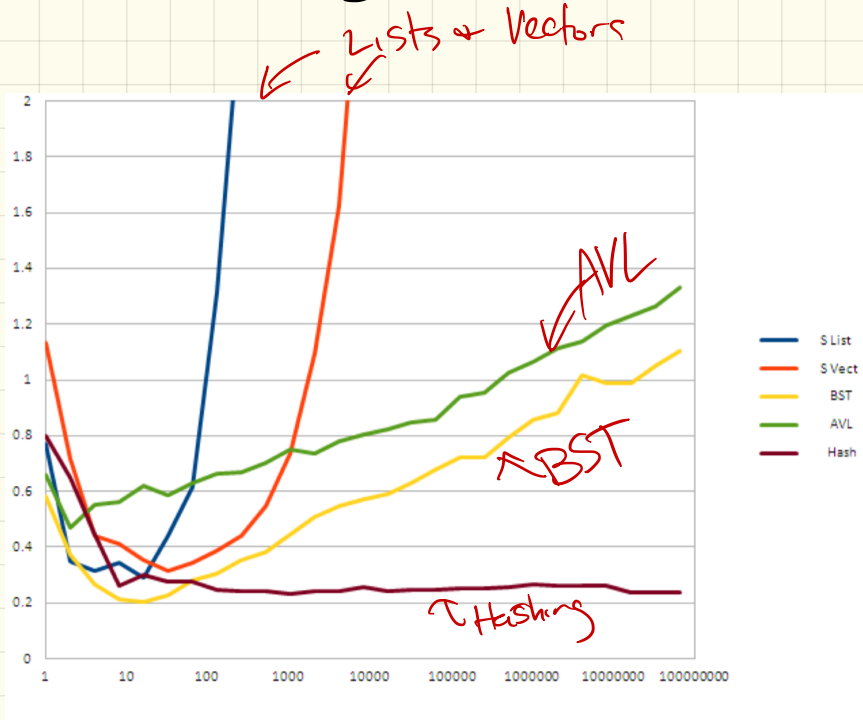
$n, n-1, n-2, \dots, 1$





# Random inserts:

Fix  $n$  of insert  $1-n$  in random order:



# Take away:

- Hashing - wow!  
Caveat: limited, don't implement yourself!
- Also - your data does matter.

Performance varies dramatically.

- These are "asymptotic",  
but remember that  
constant factors can  
still be meaningful.

Now:

Thanks for a lovely  
(if busy!) semester!  
I hope to see you  
all around next year.

Questions: about the final,  
or next week?

(+ finally - evaluations!!  
(you have time!))