CS2100

Overview & Recap

Final announcements

- HW + lab - due today/tomorrow
- Review session Monday
  - bring questions!!
- Final: Wed. at ~~8am~~ 2pm, here
- Keep an eye on blackboard/git for more grading
- Request: instructor evals!
  (You'll have time at end)

- Practice finals -
  come by my office

Data Structures we've seen:

- Stacks
- queues

] O(1) always

- Lists
- Vectors

- Trees
- Binary Trees
- (Balanced) BSTs

50-60%

- Heaps
- Huffman Trees
- Treaps
- Hashing
- Graphs
- Sets

Also:

- C++ - tons! ←
- Sorting / searching ←

Trade-offs:

Simple + limited:

- stacks
- queues
- even priority queues

- hashing ??

Why use?

Speed

(don't build functionality if user shouldn't have it)

"Full-featured":

- Vectors
- Lists
- Trees
- Hashing?

O(1) vs O(n)

O(log n)

O(1) (mostly)

Trade-offs are key!

Consider:

- your data
- & how you'll use it

Practical vs theoretical

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

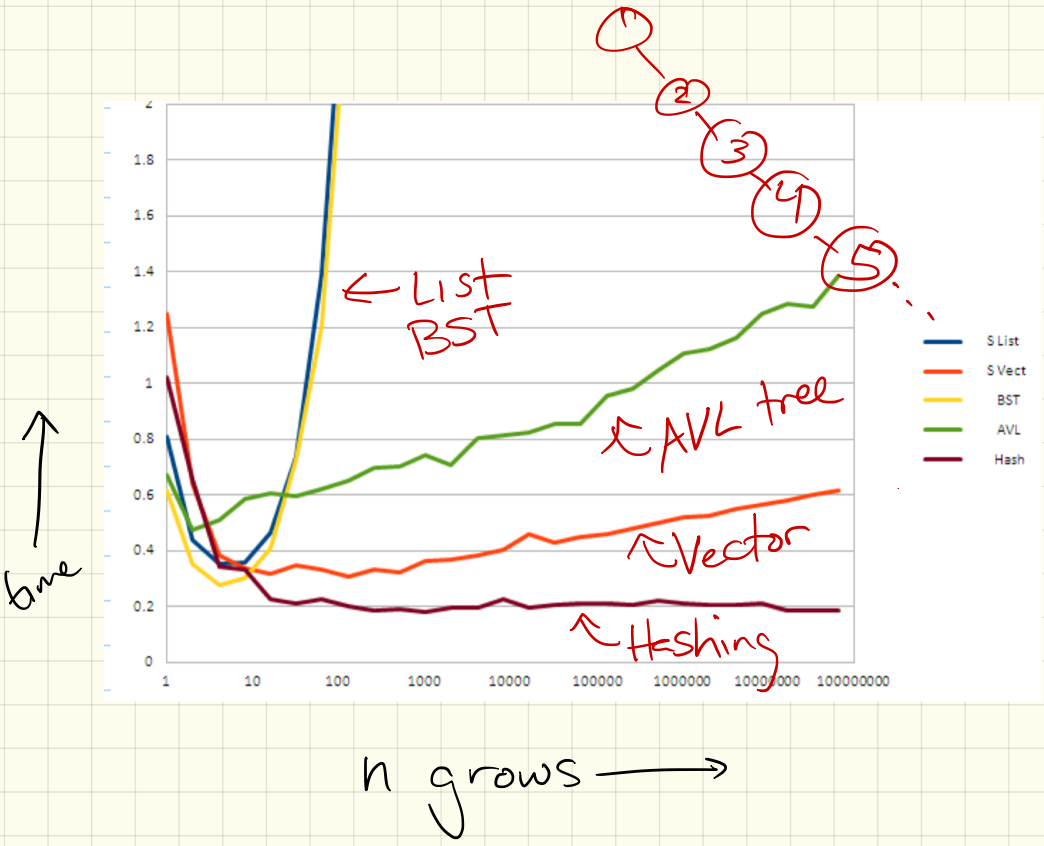
(depending on data)

- hashing
- quicksort
- even inserting in a vector (well - push_back)

Some data:

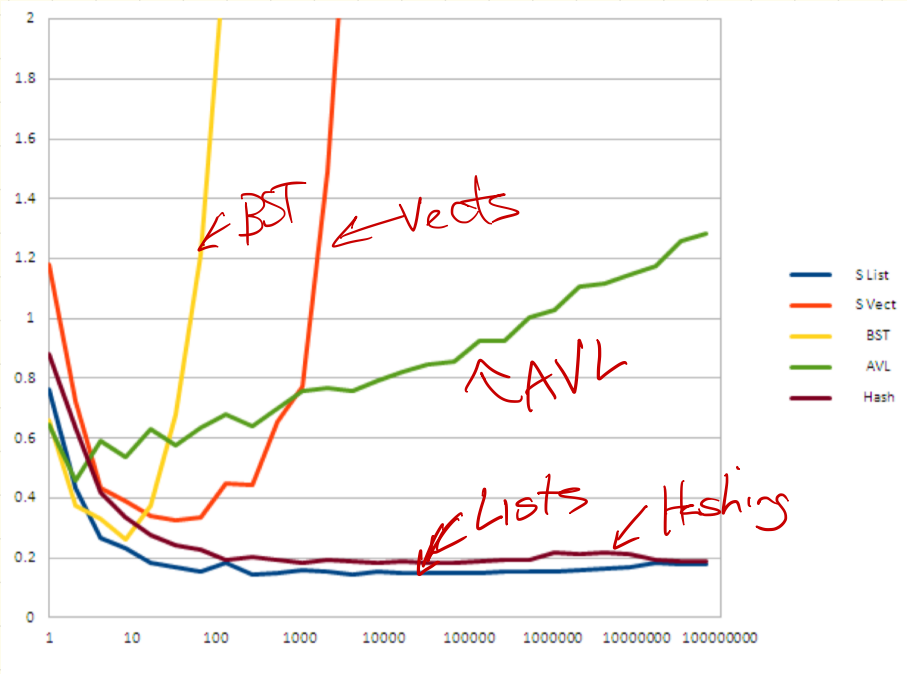
Insertions done in-order:

1, 2, 3, ..., n



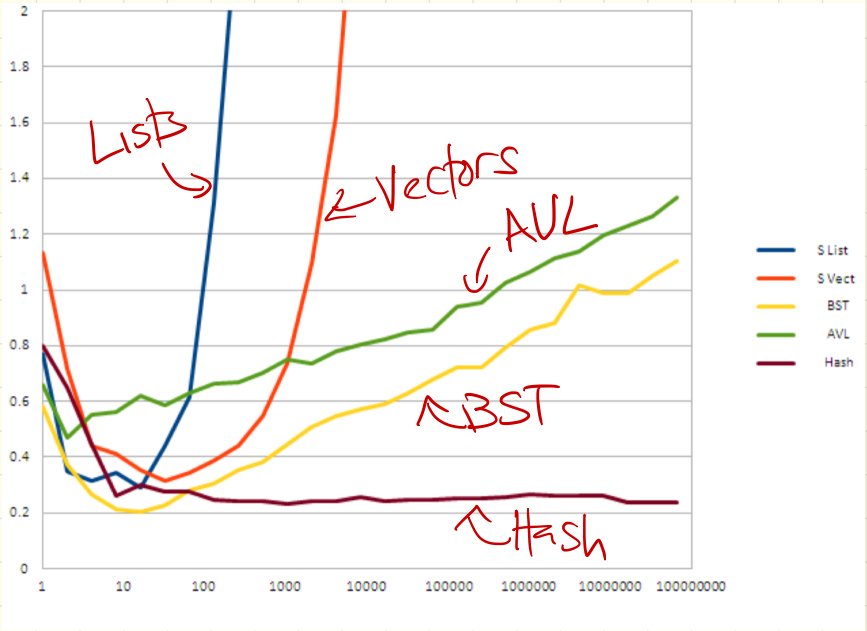
Reverse order inserts

$n, n-1, n-2, \ldots, 1$



Random inserts:

Fix n & insert 1-n in random order:



Take away:

- Hashing – wow! :)

  Warning: May not want to trust all implementations.

- Also – your data does matter.

  Performance varies drastically.

- 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!))