CS 2100 - Recap of our semester

Note Title 12/12/2011

Announcements

- HW due now
- Practice final is at front of the room
- Review session: Monday during last class
- Final exam: Wednesday at 8am

Data Structures Covered

- stacks
- queues
- lists
- vectors
  - → sorting
  - → searching
- trees:
  - BST
  - AVL Trees (balanced binary trees)
  - Huffman trees
  - treaps
  - heaps
- hashing
- graphs

Data Structures

Some data structures have limited functionality, but as a result are extremely efficient.

Ex.

- stack
- queue

"Full-featured" data structures

More versatile data structures have trade-offs:
to get something faster, you sacrifice speed in another area.

Ex:

- list, vector: insert/remove vs access time
- AVL graphs: space vs access time

Randomized or Expected

Some work well in practice, but have no theoretical guarantees.
↑ worst case

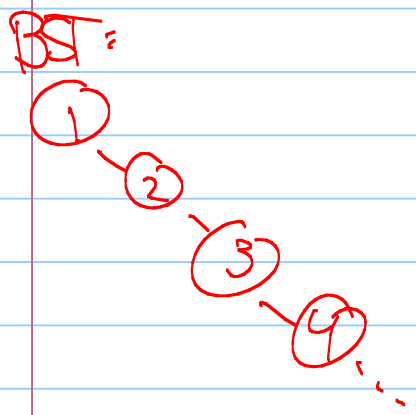
Ex:

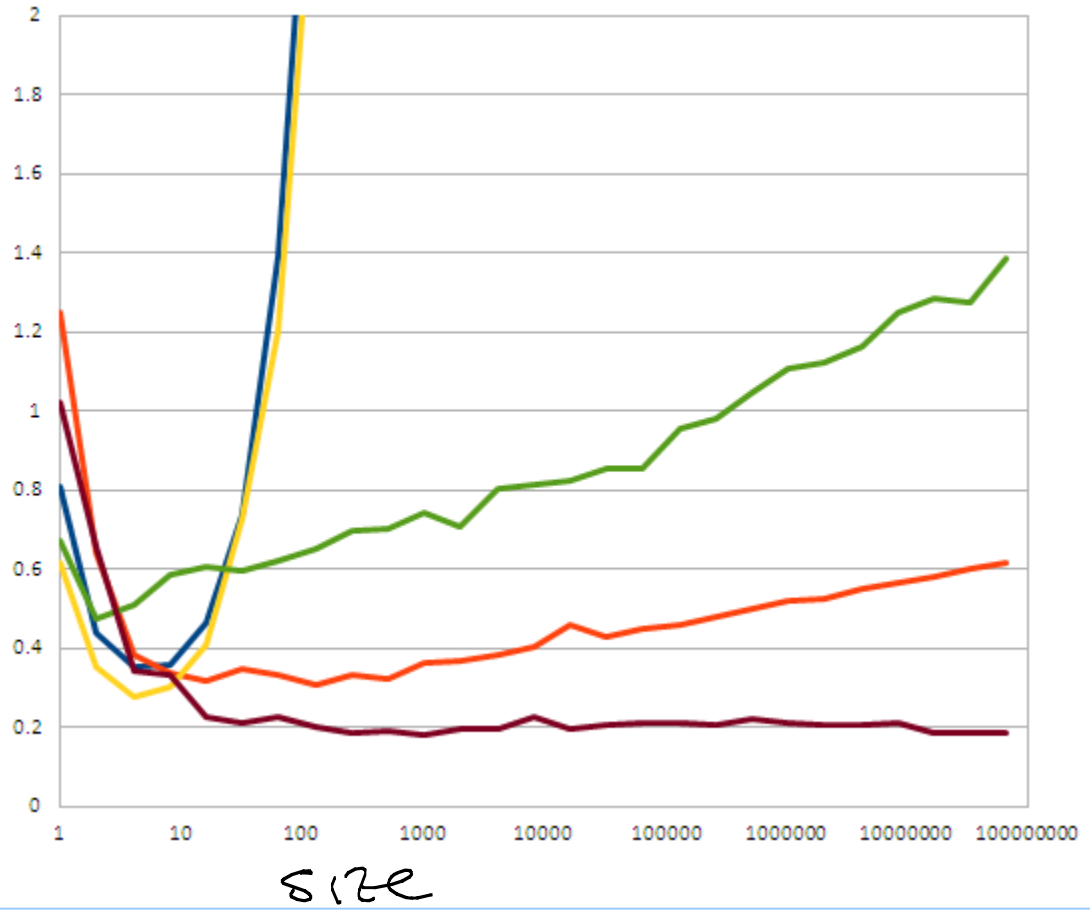
- hash functions
- quick sort
- amortized push_back in vector

So which is best?

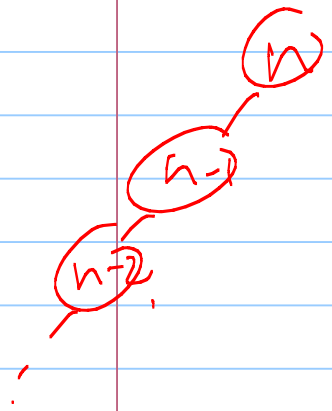
Ans: Depends!

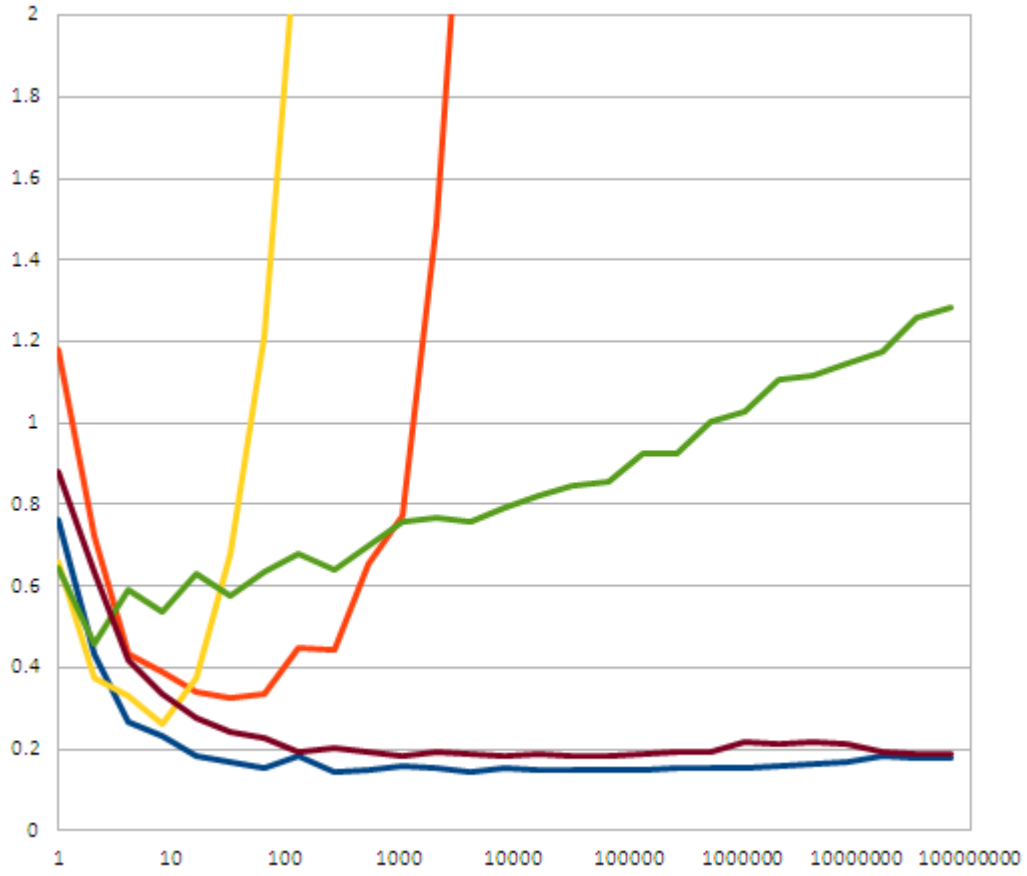
In-order insertions: insert: 1, 2, 3, 4, ..., n



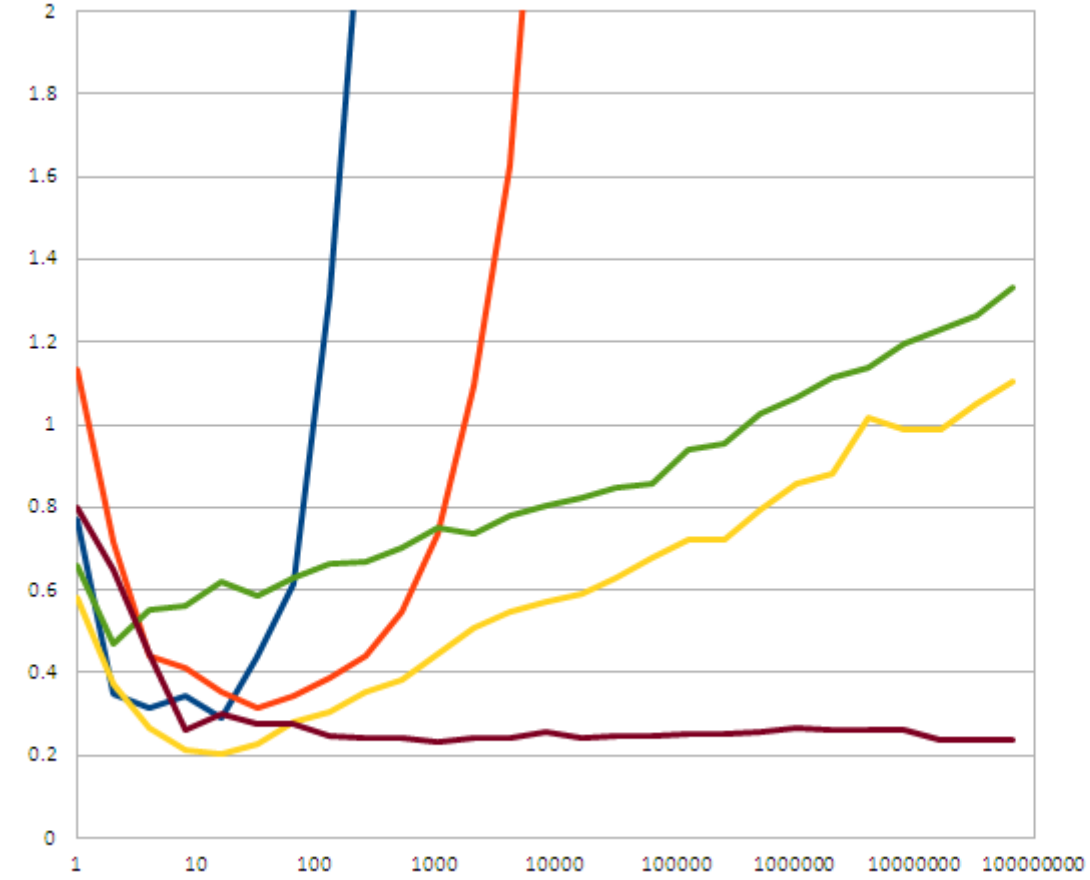


Reverse Order inserts : n, n-1, n-2, ..., 1





Random Inserts



Note: Hashing is fast!

The "randomness" of the hash function even hides order of elements.

Caution, though: They don't have all the extra functionality of others.