Efficiency (time)

- operations on BSTs
  → efficiency depends on height.
- best vs worst case → on tree with n nodes
  - worst case for BST search?
    - very unbalanced
    - nearly a Linked List
    - $\Theta(n)$ → time proportional to # of nodes.
  - worst case for search in balanced BST?

$$\Theta(\lg n)$$

  - best case (for either)
    - what we're looking for is the root $\Theta(1)$

best/worst case NOT $\overbrace{\text{size of input}}^{n}$

How to maintain a balanced tree?

- more work to maintain balance. → CSC263
- is it worth it to balance?

→ average case analysis

SORTING!!!

- understand the algorithms
- understand how to analyze
- never implement your own*

3 algorithms
- iterative → loops in loops
- all $\Theta(n^2)$
- common:

$$1 + 2 + \ldots + (n-1) + n \Rightarrow \frac{n(n+1)}{2} \Rightarrow \Theta(n^2)$$

We can do better with recursion

Divide + Conquer

1. divide into smaller problems
2. solve subproblems recursively
3. combine to get result.

Mergesort, Quicksort
- Quicksort → more dividing work
- Mergesort → more combining work