Exam 1 Review:

Name	Best	Average	Worst	Memory	Stable
Quick Sort	$n \log(n)$	$n \log(n)$	n^2	$\log(n)$	No
Merge Sort	$n \log(n)$	$n \log(n)$	$n \log(n)$	n	Yes
Selection Sort	n^2	n^2	n^2	1	No
Insertion Sort	n	n^2	n^2	1	Yes
Heap Sort	$n \log(n)$	$n \log(n)$	$n \log(n)$	1	No

- Least # of steps for an algorithm to input array in reverse order (largest to smallest)?
  ↳ Merge Sort
- Least # of steps for an algorithm to input nearly sorted array?
  ↳ Insertion Sort

- Array [9, 4, 5, 1, 5, 2]

Insertion - Stable!

1. [9 | 4, 5, 1, 5, 2]
2. [4, 9 | 5, 1, 5, 2]
3. [4, 5, 9 | 1, 5, 2]
4. [1, 4, 5, 9 | 5, 2]
5. [1, 4, 5, 5, 9 | 2]
6. [1, 2, 4, 5, 5, 9] END

Selection - Not Stable!

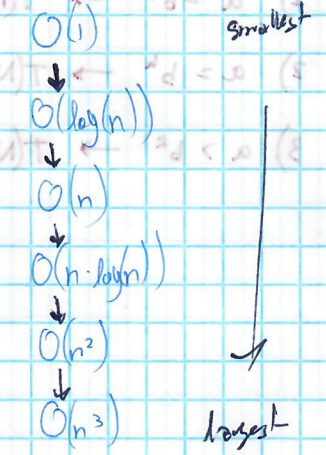
1. [1 | 9, 4, 5, 5, 2]
2. [1, 2 | 9, 4, 5, 5]
3. [1, 2, 4 | 9, 5, 5]
4. [1, 2, 4, 5 | 9, 5]
5. [1, 2, 4, 5, 5 | 9]
6. [1, 2, 4, 5, 5, 9] END

Big-O: $f \in O(g) \iff f(n) \leq k \cdot g(n)$ as $n \to \infty$
↳ $4n^3 + 3n - 5 \in O(n^3) \Rightarrow$ printPairs $\in O(n^2)$

Big-Ω: $f \in \Omega(g) \iff g \in O(f) = g(n) \leq k \cdot f(n)$ as $n \to \infty$
↳ "lower bound"

Big-Θ: $f \in \Theta(g) \iff f \in O(g)$ & $f \in \Omega(g)$
$k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n)$ as $n \to \infty$
↳ "tight bound"

Growth



Quick Sort:

- last element in list (pivot)
- put all elements into 2 lists
  - all elements ≤ pivot ;
  - all elements > pivot ;
- recursively call function until sorting is corrected
- left list + pivot + right list

Selection Sort:

- first element = smallest element
- second element will be compared to first
- If first > second, swapped

Min Heap: tree element ≥ parent

Max Heap: tree element ≤ parent
- add to end of heap & swap w/ parent (Bubbling Up)
- peek at top element
- take top element, return it, swap last element in heap & swap that element down with children

Recurrence Relation: $T(n) = a\,T(n/b) + f(n)$

$a > 0$ — # of recursive calls made at each step?

$b > 1$ — reducing problem by what multiplicative factor?

$f(n)$ — # of work needed to be done @ recursive call?

Master Theorem: given ↑ $T(n) = a\,T(n/b) + \Theta(n^k)$, one of the following is true

1) $a < b^k \rightarrow T(N) \in \Theta(n^k)$
2) $a = b^k \rightarrow T(N) \in \Theta(n^k \log(n))$
3) $a > b^k \rightarrow T(N) \in \Theta(n^{\log_b a})$

Merge Sort:

- divide list into left & right halves
  - recursively sort the halves
- merge list back together into one big sorted list
- merge (left, right)

Insertion Sort:

- first element = smallest element
- second element compared to first = next element in list
- compare elements & insert by shifting all other elements to the right by one
- increment to next element