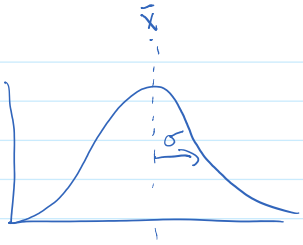
Sorting (Insertion, Quick, ...)

Tuesday, October 20, 2015 11:33 AM

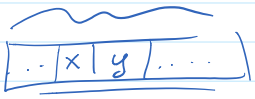
Written Exam Stats

$\bar{x} = 71$
$\tilde{x} = 75$
$\sigma = 24$



Insertion sort

if sorted ⇒ inner loop doesn't occur



in total, insertion sort $O(n)$

$O(n \cdot k)$ — n

```
for ( int i=0; i < list.size(); i++ ) {
    int j = i; //O(1)
    while ( j > 0 && list.get(j) < list.get(j-1) ) {
        swap ( list, j, j-1 ); //O(1)
        j--;
    }
}
```

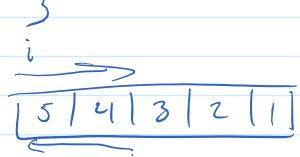
Aside Need an $O(1)$ op. to change elements w/in our list.

```
ArrayList<E>{
    E[] theArray;
    ⋮
    void set (int index, E elem) {
        theArray[index] = elem;    O(1)
    }
}
```

assume integers

```
public static void swap ( list, int index1, int index2 ) {
    int temp = list.get(index1); //O(1)
    list.set(index1, list.get(index2)); //O(1)
    list.set(index2, temp);    // O(1)
}
```

$O(1)$



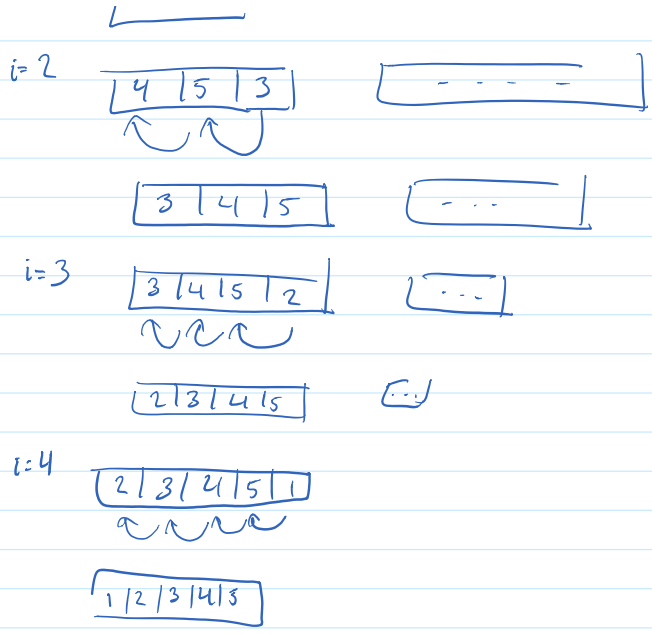
i=0

i=1

j=1

while (j>0 && list.get(j) < list.get(j-1))

T — 4 — 5




```
for (int i = 0; i < n; i++) {
    constant();
    for (j = i; j > 0; j--) {
        constant();
    }
}
```

$O(n^2)$ "Worst case" insertion sort

Insertion sort

$\Omega(n)$	best
$O(n^2)$	worst
$O(n^2)$	avg

Divide & Conquer / Conquer & Divide algorithms

Quicksort

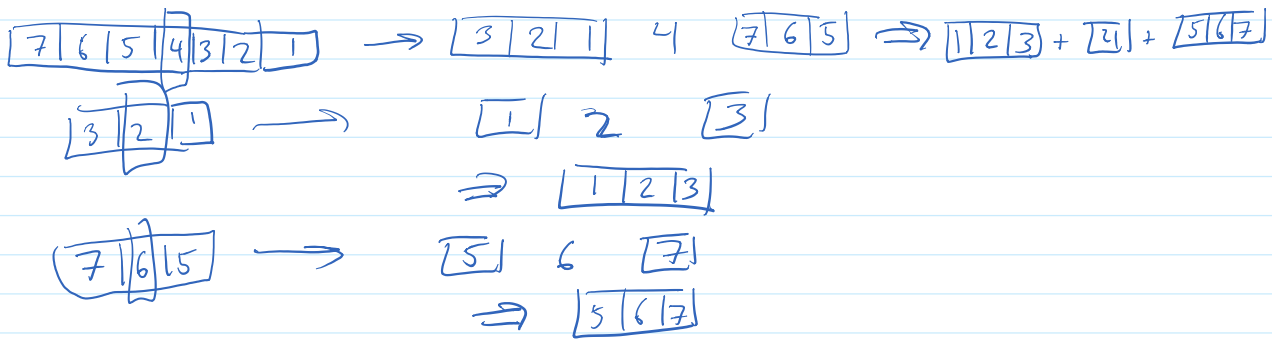
```
fun quicksort (list):
    choose a "pivot" from the list.   // needs clarification
    (left, right) = partition (list, pivot)   O(n)
    quicksort (left)   // Quicksort (n/2) (best case)
    quicksort (right)  // Quicksort (n/2)
    return left + [pivot] + right
endfunction

fun partition (list, element):
    left = empty list
    right = empty list
    for item in list:
        if item < element:
            left.append (item)
        else:
```

$O(n)$

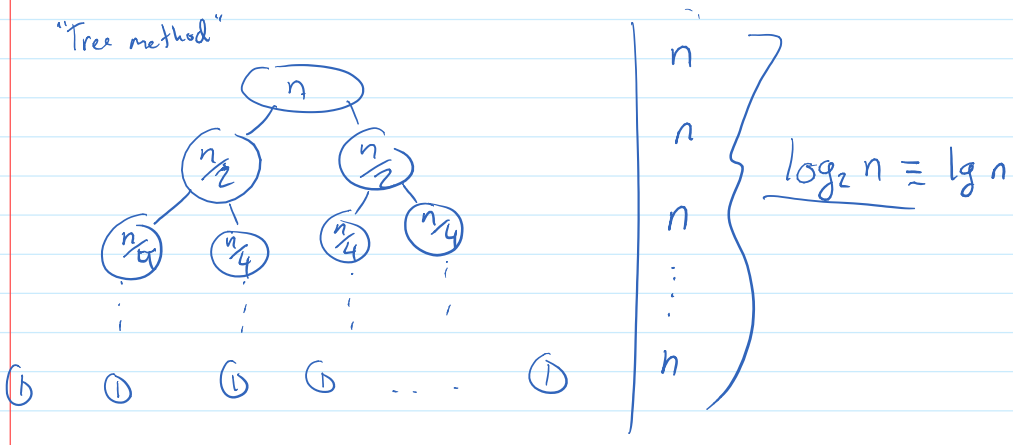
```
            left.append(item)
        else:
            right.append(item)
        endif
    endfor
    return (left, right)
endfunction
```

$O(n)$



Best possible pivot splits the list in half

"Tree method"



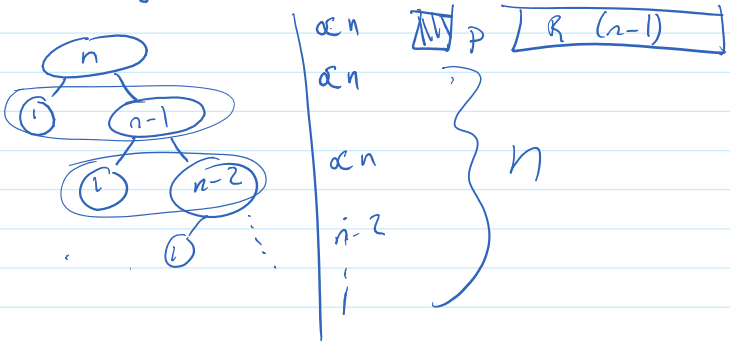
$2^i = n$

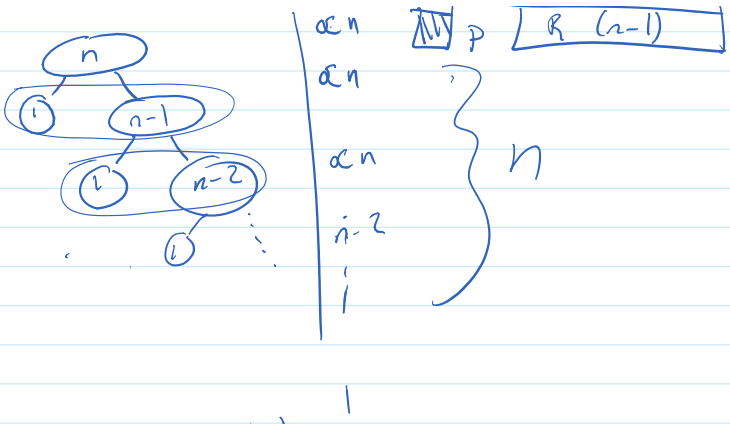
$\log_2 2^i = \log_2 n$

$i = \log_2 n$

$\Rightarrow$ best case $\Omega(n \lg n) \Rightarrow$ bigger than $\Omega(n)$

Worst case quick sort

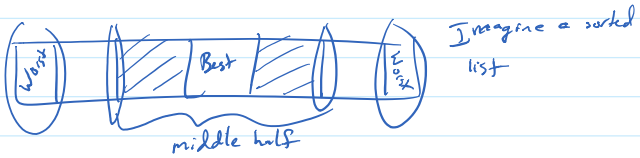




Worst case = $O(n^2)$

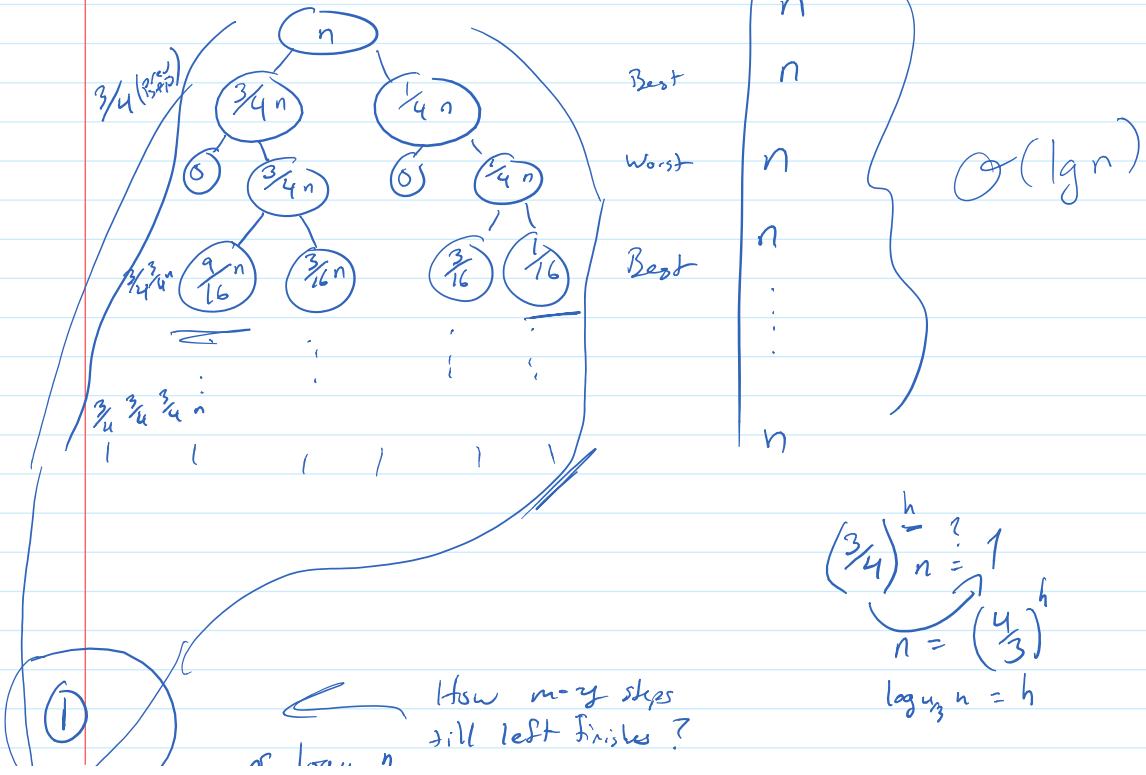
	Insertion	Quick
Best	$\Omega(n)$	$\Omega(n \lg n)$
Worst	$O(n^2)$	$O(n^2)$
Avg	$O(n^2)$	?

Avg Quicksort



Imagine a sorted list

Choosing a pivot randomly 50% of the time will be from the middle half



$O(\lg n)$

$\left(\frac{3}{4}\right)^h n = 1$?

$n = \left(\frac{4}{3}\right)^h$

$\log_{4/3} n = h$

(1)

How many steps till left finishes?

$\log_{4/3} n = h$

$\propto \log_{4/3} n$

Avg $\Rightarrow$ $O(n \lg n)$

	Insertion	Quick
Best	n	$n \lg n$
Worst	n^2	n^2
Avg	n^2	$n \lg n$

Merge Sort
Heap Sort