Topics Covered Topic 1 - 8

Topic 2 Insertion Sort

```
INSERTION-SORT(A)
1  for j = 2 to A.length
2      key = A[j]
3      // Insert A[j] into the sorted sequence A[1 .. j − 1].
4      i = j − 1
5      while i > 0 and A[i] > key
6          A[i + 1] = A[i]
7          i = i − 1
8      A[i + 1] = key
```

- Best Case runtime: $\Theta(n)$
- Worst Case Run-Time: $\Theta(n^2)$

Loop Invariants

A formal property that is true at the start of each iteration.

- Initialization : True prior to the first iteration
- Maintenance: true prior to the given iteration, then it remains true before the next iteration
- Termination: loop terminates and gives a useful property to show that the algorithm is correct

Divide and Conquer

- Divide: the problem into sub smaller instances
- Conquer: solve the problems recursively
- Combine: subproblem into the solution of the original

Merge-Sort

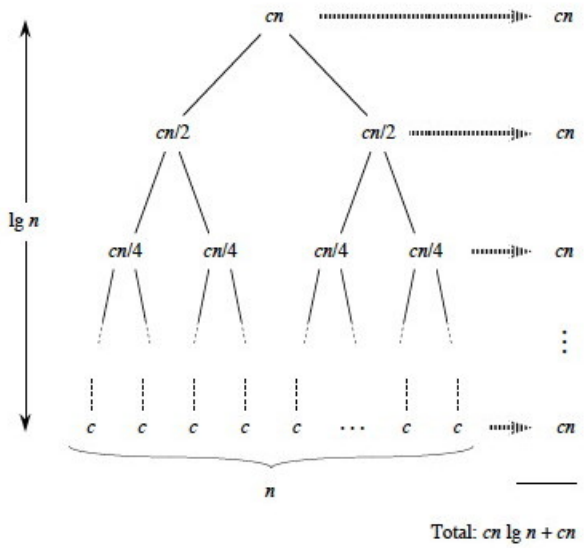
```
MERGE-SORT(A, p, r)
1  if p < r
2      q = ⌊(p + r)/2⌋
3      MERGE-SORT(A, p, q)
4      MERGE-SORT(A, q + 1, r)
5      MERGE(A, p, q, r)
```

- Best Case: $\Theta(n \log n)$

- Worst Case: $\Theta(n \log n)$

Recursion Tree Analysis

$$T(n) = c \quad \text{if} \quad n = 1$$
$$= 2T(\frac{n}{2}) + cn \quad \text{if} \quad n > 1$$



Topic 3

Algorithm	Worst-case running time	Average-case/expected running time
Insertion sort	$\Theta(n^2)$	$\Theta(n^2)$
Merge sort	$\Theta(n \lg n)$	$\Theta(n \lg n)$
Heapsort	$O(n \lg n)$	—
Quicksort	$\Theta(n^2)$	$\Theta(n \lg n)$ (expected)
Counting sort	$\Theta(k + n)$	$\Theta(k + n)$
Radix sort	$\Theta(d(n + k))$	$\Theta(d(n + k))$
Bucket sort	$\Theta(n^2)$	$\Theta(n)$ (average-case)

Asymptotic Bounds

- O $\approx \leq$, upper bound, worst case
- $\Omega \approx \geq$, best case, lower bound
- $\Theta \approx =$, no better or no worse
- o $\approx <$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

- $\omega \approx >$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

- **theorem 3.1** $\mathrm{f(n)} = \Theta(g(n))$ iff $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

Transitivity:

- $f(n) = \Theta(g(n))$ and $g(n) = \Theta(\mathrm{h(n)}) \Rightarrow f(n) = \Theta(\mathrm{h(n)})$.
- $f(n) = \mathrm{O}(g(n))$ and $g(n) = \mathrm{O}(\mathrm{h(n)}) \Rightarrow f(n) = \mathrm{O}(\mathrm{h(n)})$.
- $f(n) = \Omega(g(n))$ and $g(n) = \Omega(\mathrm{h(n)}) \Rightarrow f(n) = \Omega(\mathrm{h(n)})$.
- $f(n) = \mathrm{o}(g(n))$ and $g(n) = \mathrm{o}(\mathrm{h(n)}) \Rightarrow f(n) = \mathrm{o}(\mathrm{h(n)})$.
- $f(n) = \omega(g(n))$ and $g(n) = \omega(\mathrm{h(n)}) \Rightarrow f(n) = \omega(\mathrm{h(n)})$.

Reflexivity:

- $f(n) = \Theta(f(n))$
- $f(n) = \mathrm{O}(f(n))$
- $f(n) = \Omega(f(n))$
- *What about o and ω?*

Symmetry:

- $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$
- *Should any others be here? Why or why not?*

Transpose Symmetry:

- $f(n) = \mathrm{O}(g(n))$ iff $g(n) = \Omega(f(n))$
- $f(n) = \mathrm{o}(g(n))$ iff $g(n) = \omega(f(n))$