L18: Complexity and Algorithm Analysis

Comparing Algorithms

$5n + 6 = ? \quad 10n + 2 = ? \quad 2n + 50$

- Growth rates & Asymptotic bound ($n \to \infty$)
- Alg. A vs. alg. B: which algorithm has the smaller asymptotic time bound?

Computational Complexity

- The term complexity of an algorithm refers to its asymptotic bound. That is, the order of that bound.

May be better for small $n$, but as soon as $n$ increases, the algorithm becomes inefficient.
“Big-O” and “Big-Θ” notation

**Definition 1:** A function $f(n)$ is said to be “order of $g(n)$” or “big-O of $g(n)$” if

$\exists N_o, c > 0$ such that $f(n) \leq cg(n)$, for $n \geq N_o$.

**Notation:** $f(n) = O(g(n))$

**Definition 2:** A function $f(n)$ is said to be “big-Θ of $g(n)$” if

$\exists N_o, c_1, c_2 > 0$ s. t. $f(n) \leq c_1g(n)$ and $f(n) \geq c_2g(n)$, for $n \geq N_o$.

**Notation:** $f(n) = \Theta(g(n))$
Examples
Selection Sort

\[ a[ ] = 15, \ 6, \ 18, \ 3, \ 7, \ 17, \ 20, \ 1, \ 4, \ 13, \ 9 \]
L18: Complexity and Algorithm Analysis

**SelectionSort** \( a[], \) lower, upper \) 

\[
\text{for}(\text{low} = \text{lower}; \ \text{low}<\text{upper}; \ \text{low}++) \ { \\
\text{smallest} = \text{findSmallest}(a, \text{low}, \text{upper}) \ \\
\text{tmp} = a[\text{low}] \ \\
\text{a}[\text{low}] = a[\text{smallest}] \ \\
\text{a}[\text{smallest}] = \text{tmp} \} \ \\
\text{return} \}
\]

**findSmallest** \( a[], \) lower, upper \) 

\[
\text{smallestIndex} = \text{lower} \ \\
\text{for}(i = \text{lower}+1; \ i\leq\text{upper}; \ i++) \ \\
\text{if}(a[i] < a[\text{smallestIndex}]) \ \\
\text{smallestIndex} = i \ \\
\text{return smallestIndex} \}
\]

\( T(n) = \), \( \Omega \)

Neglect operations that are independent of the size of the input.
L18: Complexity and Algorithm Analysis

\[ T(n) = ?, \quad \Omega ? \]
Recursive version

```plaintext
SelectionSortR(a[], lower, upper) {
    if lower = upper
        return
    smallest = findSmallest(a, low, upper)   n
    tmp = a[low]                              1
    a[low] = a[smallest]                      1
    a[smallest] = tmp                         1
    SelectionSortR(a[], lower+1, upper)      T(n-1)
}
```

\[
T(1) = 1
T(n) = T(n-1) + n + 4
\]

Recurrence: O?
Recurrences

- **Iteration method:** iterate, express as $\Sigma$, solve $\Sigma$
Assignment

• Read Sections 14.6 – 14.8

• Additional readings posted on Blackboard (QuickSort-1 and QuickSort-2)

• Homework #3