CSE 250 Recitation

February 13 - 14: Asymptotic Analysis in Code

Analyzing Code/Algorithms

Remember the different types of control flow:

- Sequential (statements executed one after another)
  - Add the number of steps together
  - If I do A then B, the total cost is the cost to do A plus the cost to do B
- Selection (conditional execution of statements)
  - Our growth function will be a piecewise function
- Repetition (repeating execution of one or more statements)
  - Add up the total number of steps…with summations

Code Analysis

```
Structure ds = new Structure();

ds.initialize();

for (int i = 0; i < n; i++) {
  Thing thing = createAThing(i, n);
  ds.insertAThing(thing);
}

if (weFeelLikeIt())
  ds.computeSomething();
```

Function	Growth Function
`Structure()`	T_{new} =
`ds.initialize()`	T_{init} =
`createAThing()`	T_{create} =
`ds.insertAThing()`	T_{insert} =
`weFeelLikeIt()`	T_{cond} =
`ds.computeSomthing()`	T_{comp} =

Exercise: Write the growth function, ***T*(*n*)**, that represents the runtime of this code

(start by writing it in terms of the other growth functions)

Growth Function

Here is the growth function you should have gotten

(If T_{cond} was pulled out of the piecewise that's fine too)

```java
Structure ds = new Structure();

ds.initialize();

for (int i = 0; i < n; i++) {
  Thing thing = createAThing(i, n);
  ds.insertAThing(thing);
}

if (weFeelLikeIt())
  ds.computeSomething();
```

$$T(n) = T_{new} + T_{init} + \sum_{i=0}^{n-1} (T_{create} + T_{insert}) + \begin{cases} T_{cond} + T_{comp} & \text{if weFeelLikeIt()} \\ T_{cond} & \text{otherwise} \end{cases}$$

Finding Bounds

Exercise:

1. Update your growth function with the growth functions to the right
2. Find its closed form solution
3. Determine the $\boldsymbol{O}$, $\boldsymbol{\Omega}$, and $\boldsymbol{\Theta}$ bounds

Note: $|x|$ is the notation for "size of x"

You may assume that **`insertAThing`** increases the size of the data structure by one, and nothing else changes its size

Function	Growth Function
`Structure()`	$T_{new} = 5$
`ds.initialize()`	$T_{init} = 23$
`createAThing()`	$T_{create} = 6$
`ds.insertAThing()`	$T_{insert} = 2 \cdot \lvert ds \rvert$
`weFeelLikeIt()`	$T_{cond} = 4n + 5$
`ds.computeSomthing()`	$T_{comp} = 3 \cdot \lvert ds \rvert^3$

Getting the Closed Form Solution

$$T(n) = 5 + 23 + \sum_{i=0}^{n-1}(6 + 2i) + \begin{cases} 4n + 5 + 3n^3 & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + 3n^3 & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

Finding Bounds (Big-O)

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + 3n^3 & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

$$O(1) + O(n^2) + O(n^3) = O(1 + n^2 + n^3) = O(n^3)$$

Finding Bounds (Big-Ω)

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + 3n^3 & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

$$\Omega(1) + \Omega(n^2) + \Omega(n) = \Omega(1 + n^2 + n) = \Omega(n^2)$$

Finding Bounds (Big-Θ)

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + 3n^3 & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

$$T(n) \in O(n^3)$$

$$T(n) \in \Omega(n^2)$$

Since both of these bounds are **tight** then there does not exist an f(n) such that

$$T(n) \in \Theta(f(n))$$

Finding Bounds (Big-Θ)

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + \boxed{3n^3} & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

Follow Up: What if T_{comp} was **$3n^2$** instead? What if it was **3n**?

Finding Bounds (Big-Θ)

$$T(n) = 28 + (5n + n^2) + \begin{cases} 4n + 5 + \boxed{3n^3} & \text{if weFeelLikeIt()} \\ 4n + 5 & \text{otherwise} \end{cases}$$

Follow Up: What if T_{comp} was **$3n^2$** instead? What if it was **3n**?

In both cases, T(n) would be in ***O*(n^2)**, **Ω(n^2)** and therefore **Θ(n^2)**

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First…what is the definition of big-O?

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First...what is the definition of big-O?

$$g(n) \leq c\, f(n), \text{ for all } n \geq n_0 \text{ for some } c > 0, \text{ and } n_0 \geq 0$$

What is the definition of big-Ω?

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First...what is the definition of big-O?

$$g(n) \leq c\, f(n), \text{ for all } n \geq n_0 \text{ for some } c > 0, \text{ and } n_0 \geq 0$$

What is the definition of big-Ω?

$$g(n) \geq c\, f(n), \text{ for all } n \geq n_0 \text{ for some } c > 0, \text{ and } n_0 \geq 0$$

Proving Bounds

Exercise: Prove the following

$$g(n) = 3n + n^2$$

1. $g(n) \in O(n^2)$
2. $g(n) \in \Omega(n^2)$
3. $T(n) \in O(n^3)$
4. $T(n) \in \Omega(n^2)$

$$T(n) = n^2 + 5n + 28 + \begin{cases} 3n^3 + 4n + 5 & \text{if true} \\ 4n + 5 & \text{otherwise} \end{cases}$$

Hint: For T, consider the fact that it is either:

$$3n^3 + n^2 + 9n + 33 \quad \text{OR} \quad n^2 + 9n + 33$$

More Examples

Prove the following:

$12 \log(10 \times 2^n) \in O(n)$

$n^2 + n \log(n) \in O(2^n)$

$n^2 + 15n^3 \in \Omega(n)$

$$\sum_{i=1}^{n} i \in \Omega(n^2)$$