Slide 2

Big $O$ and Big $\Theta$

Recall the definitions:

Let $f(x)$ and $g(x)$ be two functions.

We say that $f(x)$ is $O(g(x))$ or $f(x) = O(g(x))$ if there are numbers $M$ and $C$ such that

$$|f(x)| \leq C|g(x)| \quad \text{for } x \geq M$$

We say that $f(x)$ is $\Theta(g(x))$ or $f(x) = \Theta(g(x))$ if there are numbers $C,D$ and $M$ such that

$$D|g(x)| \leq |f(x)| \leq C|g(x)| \quad \text{for } x \geq M$$
Relations Between Big $O$ and Big $\Theta$

1. $f(x) = \Theta(g(x))$ if and only if $f(x) = O(g(x))$ and $g(x) = O(f(x))$.

(a) Suppose $f(x) = \Theta(g(x))$. We will show that $f(x) = O(g(x))$ and $g(x) = O(f(x))$.

Since $f(x) = \Theta(g(x))$ there are numbers $C,D$ and $M$ such that

$$D|g(x)| \leq |f(x)| \leq C|g(x)| \quad \text{for} \quad x \geq M$$

The inequality $|f(x)| \leq C|g(x)|$ says that $f(x) = O(g(x))$.

The inequality $D|g(x)| \leq |f(x)|$ can be rewritten

$$|g(x)| \leq \frac{1}{D}|f(x)|$$

which says that $g(x) = O(f(x))$. 

(b) Now suppose that \( f(x) = O(g(x)) \) and \( g(x) = O(f(x)) \). We will show that \( f(x) = \Theta(g(x)) \).

First if \( f(x) = O(g(x)) \) the there are numbers \( C \) and \( M \) such that
\[
|f(x)| \leq C|g(x)|
\]
for \( x \geq M \). If \( g(x) = O(f(x)) \) the there are numbers \( D \) and \( N \) such that
\[
|g(x)| \leq D|f(x)|
\]
for \( x \geq N \).

This implies
\[
\frac{1}{D}|g(x)| \leq |f(x)|
\]
for \( x \geq N \).

Thus for \( x \geq \max(M, N) \)
\[
\frac{1}{D}|g(x)| \leq |f(x)| \leq C|g(x)|
\]
which is just the requirement that \( f(x) = \Theta(g(x)) \).
2. If $f(x) = \Theta(g(x))$ then $g(x) = \Theta(f(x))$.

Suppose $f(x) = \Theta(g(x))$. We will show that $g(x) = \Theta(f(x))$.

Since $f(x) = \Theta(g(x))$ there are numbers $C,D$ and $M$ such that

$$D|g(x)| \leq |f(x)| \leq C|g(x)|$$

for $x \geq M$

---

(a) From the inequality

$$D|g(x)| \leq |f(x)|$$

we deduce

$$|g(x)| \leq \frac{1}{D}|f(x)|$$

(b) From the inequality

$$|f(x)| \leq C|g(x)|$$

we deduce

$$\frac{1}{C}|f(x)| \leq |g(x)|$$
Putting these inequalities together we get

\[
\frac{1}{C}|f(x)| \leq |g(x)| \leq \frac{1}{D}|f(x)|
\]

with both holding for \( x \geq M \).

This says that \( g(x) = \Theta(f(x)) \).

Similarly, if \( g(x) = \Theta(f(x)) \) then \( f(x) = \Theta(g(x)) \).

So \( f(x) = \Theta(g(x)) \) is exactly the same thing as \( g(x) = \Theta(f(x)) \).

---

**The Meaning of \( \Theta \) Notation**

In the previous lecture we noted that \( f(x) = \Theta(g(x)) \) means that the absolute value of \( f(x) \) is “sandwiched” between multiples of the absolute value of \( g(x) \) for large enough \( x \).

Since \( f(x) = \Theta(g(x)) \) is equivalent to \( g(x) = \Theta(f(x)) \) it follows that the absolute value of \( g(x) \) is “sandwiched” between multiples of the absolute value of \( f(x) \) for large enough \( x \).
Example

In the previous lecture we saw that if
\[ f(x) = \frac{3\sqrt{x}(2x + 5)}{|x| + 1} \quad \text{and} \quad g(x) = \sqrt{x} \]
then \( f(x) = \Theta(g(x)) \).

More precisely
\[ 3|g(x)| \leq |f(x)| \leq 9|g(x)| \]
for \( x \geq 5 \).
We also have $g(x) = \Theta(f(x))$.

In fact we see from the previous inequalities, following our proof that $f(x) = \Theta(g(x))$ is equivalent to $g(x) = \Theta(f(x))$, that

$$\frac{1}{9}|f(x)| \leq |g(x)| \leq \frac{1}{3}|f(x)|$$

for $x \geq 5$. 

---

![Plot of functions](attachment:plot.png)

**Slide 14**
When $f(x) = \Theta(g(x))$ then also $g(x) = \Theta(f(x))$. The absolute value of $f(x)$ lies between multiples of the absolute value of $g(x)$ and the absolute value of $g(x)$ lies between multiples of the absolute value of $f(x)$ all for large enough $x$.

Another way of interpreting this is that $f(x)$ and $g(x)$ are roughly proportional for large enough $x$ or that $f(x)$ and $g(x)$ grow proportionally for large $x$.

Properties of $\Theta$

1. $c \cdot f(x) = \Theta(f(x))$ for any constant $c$.
2. If $f(x) = \Theta(g(x))$ and $g(x) = \Theta(h(x))$ then $f(x) = \Theta(h(x))$

Proofs of these properties are very similar to the proofs of corresponding properties of big $O$ given in the previous lecture.
Example

We saw earlier that the number of arithmetic operations needed to evaluate a polynomial of degree $n$ are

1. $3n - 1$ operations for the usual method.
2. $2n$ operations for Horner’s method.

Since $3n - 1 = \Theta(n)$ and $2n = \Theta(n)$ it also follows that $3n - 1 = \Theta(2n)$ and vice-versa.

In other words, for large $n$ the number of operations needed to evaluate a polynomial of degree $n$ by either method is roughly proportional to $n$.

The Meaning of $O$ Notation

When $f(x) = \Theta(g(x))$ we have a pair of inequalities which “sandwich” multiples of the absolutes of the two functions between one another. From this it follows that $f(x)$ and $g(x)$ grow proportionally for large $x$.

When $f(x) = O(g(x))$ we have only one inequality

$$|f(x)| \leq C|g(x)|$$

which just says that proportionally $f(x)$ grows no faster than $g(x)$. 
If \( f(x) = O(g(x)) \) then, unlike \( \Theta \), it does not follow that \( g(x) = O(f(x)) \).

**Example**

It turns out that \( n \) is \( O(n^2) \) but \( n^2 \) is *not* \( O(n) \).

It is easy that

\[
n \leq n^2 \quad \text{for } n \geq 1
\]

so \( n = O(n^2) \).

Conversely, suppose \( n^2 = O(n) \). Then by the definition of \( O \) there would be numbers \( C \) and \( M \) such that

\[
n^2 \leq Cn \quad \text{for } n \geq M
\]

but if \( n > \max(C, M) \) then we would have \( n^2 > Cn \) contradicting the previous inequality.
Application to Algorithms

We saw in this week’s tutorial that the number of comparisons needed to sort a list of \( n \) items using the bubble sort algorithm is, in the worst case,

\[ B(n) = \frac{1}{2}(n^2 - n) \]

Now

\[ B(n) = \Theta(n^2) \]

The proof is almost the same as that for \( f(n) = \frac{1}{2}n(n + 1) \) given in the previous lecture.

A way to understand this intuitively is that for large \( n \)

\[ B(n) = \frac{1}{2}(n^2 - n) \approx \frac{1}{2}n^2 \]

This is because for large \( n \), \( n^2 \) is much larger than \( n \).

Thus, for large \( n \), \( B(n) \) is roughly proportional to \( n^2 \) which is just what \( B(n) = \Theta(n^2) \) means.

Note that this says, for example, if we increase the length of a list by a factor of 10, then we will require roughly 100 times as many comparisons to sort the list.
Big $O$ and Algorithms

What can we say about sorting a list in general. Let $S(n)$ be the number of comparisons required to sort a list of $n$ elements using the fastest algorithm possible.

All we can say is that

$$S(n) = O(n^2)$$

that is proportionally $S(n)$ grows no faster than $n^2$ (since we know the $n^2$ can be achieved by a particular algorithm, bubblesort, in the worst case).

When we know that

$$S(n) = O(n^2)$$

if we increase the length of a list by a factor of 10, then we will require no more than 100 times as many comparisons to sort the list.