Growth mindset

• **fixed mindset**: belief that you cannot increase your intelligence

• **growth mindset**: belief that you can increase your intelligence with effort
• The staff will assign student grades according to the following percentages.

• 97 A+, 93 A, 90 A-, 87 B+, 84 B, 80 B-, 76 C+, 72 C, 67 C-, 62 D+, 55 D, 50 D-, <50 F

• Exam grades might be curved to calibrate for the difficulty of the exam relative to course objectives; your curved exam grade will always be equal to or higher than your raw exam points.

TLDR; no competition except with yourself
Exam improvement

If you don’t do as well as you hoped on Midterm1, don’t fret! You have the opportunity to improve your Midterm1 score by improving on the later exams. We will use the following algorithm:

```plaintext
float avg = (midterm2.curvedscore+final.curvedscore) / 2;
if (midterm1.curvedscore < avg) {
    midterm1.curvedscore = 0.5*midterm1.curvedscore + 0.5*avg;
}
```
CS 2230
CS II: Data structures

Meeting 12: Asymptotic analysis 2
Brandon Myers
University of Iowa
Today’s big ideas

• big-Oh $O(g(n))$, big-omega $\Omega(g(n))$, and big-theta $\Theta(g(n))$ are used to describe the growth rate of functions

• can ask individually about worst, best, or average running time of an algorithm

• Try using this notation to describe the running time of some real algorithms!
Big-Oh

$R(n) \in O(g(n))$ means $g(n)$ grows same or faster than $R(n)$ up to constant factors

$\exists k > 0 . R(n) \leq k \times g(n)$ for all $n \geq n_0$

what is $n_0$? Just a sufficiently large number
Peer instruction

\[ f(n) = n^3 \log n + 10 \log n + 20n \]
\[ f(n) \in O(g(n)) \]

Pick the **best** correct \( g(n) \)

\[
\begin{align*}
    a) & \quad n^4 \\
    b) & \quad n^3 \log n + n^4 \\
    c) & \quad n^4 \log n \\
    d) & \quad 10n^3 \\
    e) & \quad n
\end{align*}
\]
Big-$\Omega$

$R(n) \in \Omega(g(n))$ means $g(n)$ grows same or slower than $R(n)$ up to constant factors

$\exists k > 0 . R(n) \geq k \times g(n)$ for all $n \geq n_0$

what is $n_0$? Just a sufficiently large number
Peer instruction

\[ f(n) = n^3 \log n + 10 \log n + 20n \]
\[ f(n) \in \Omega(g(n)) \]

Pick the best correct \( g(n) \)

\( a) \ n^4 \)
\( b) \ n^3 \log n + n^4 \)
\( c) \ n^4 \log n \)
\( d) \ 10n^3 \)
\( e) \ n \)

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+
Big-Θ

$R(n) \in \Theta(g(n))$ means $g(n)$ grows at the same rate as $R(n)$ up to constant factors

$\exists k_0, k_1 > 0 \cdot k_0 \cdot g(n) \leq R(n) \leq k_1 \cdot g(n)$ for all $n \geq n_0$

what is $n_0$? Just a sufficiently large number
Suppose we find that an algorithm has a runtime \( R(N) \) that is quadratic for the worst case input, and linear for the best case input. Which of the statements are true?

A. \( R(N) \in O(N) \)

B. \( R(N) \in O(N^2) \)

C. \( R(N) \in O(N^3) \)

D. \( R(N) \in \Theta(N) \)

E. \( R(N) \in \Theta(N^2) \)

F. \( R(N) \in \Theta(N^3) \)
Worst/Average/Best case $R(N)$

When discussing running time of an algorithm, $R(N)$, it is sometimes helpful to specify if we are interested in worst, average, or best running time.

Worst/average/best is about specific inputs **not** input size $N$.

**Do not** confuse worst/average/best with $O$, $\Theta$, $\Omega$. 
For example, in “insert random integer into sorted array” analysis we found that

**worst case** $R(N) = c \times n$ (integer goes in last spot)

All of these statements are true

\[
c \times n \in \mathcal{O}(n)
\]

\[
c \times n \in \mathcal{O}(\text{anything faster growing than } n)
\]

\[
c \times n \in \Theta(n)
\]

Likewise,

**best case** $R(N) = c$ (integer goes in first spot)

All of these statements are true

\[
c \in \mathcal{O}(1)
\]

\[
c \in \mathcal{O}(\text{anything faster growing than } 1)
\]

\[
c \in \Theta(1)
\]
Suppose we find that an algorithm has a runtime $R(N)$ that is quadratic for the worst case input, and linear for the best case input. Which of the statements are true?

A. worst case $R(N) \in O(N)$
B. worst case $R(N) \in O(N^2)$
C. worst case $R(N) \in O(N^3)$
D. worst case $R(N) \in \Theta(N)$
E. worst case $R(N) \in \Theta(N^2)$
F. worst case $R(N) \in \Theta(N^3)$
G. best case $R(N) \in O(N)$
H. best case $R(N) \in O(N^2)$
I. best case $R(N) \in O(N^3)$
J. best case $R(N) \in \Theta(N)$
K. best case $R(N) \in \Theta(N^2)$
L. best case $R(N) \in \Theta(N^3)$
Acknowledgements

• these slides borrow ideas from
  • http://homepage.divms.uiowa.edu/~ghosh/2116.html
  • http://datastructur.es/sp16/