CSCI 2824: Discrete Structures

Lecture 19: Complexity & Matrix Operations

Rachel Cox

Department of Computer Science

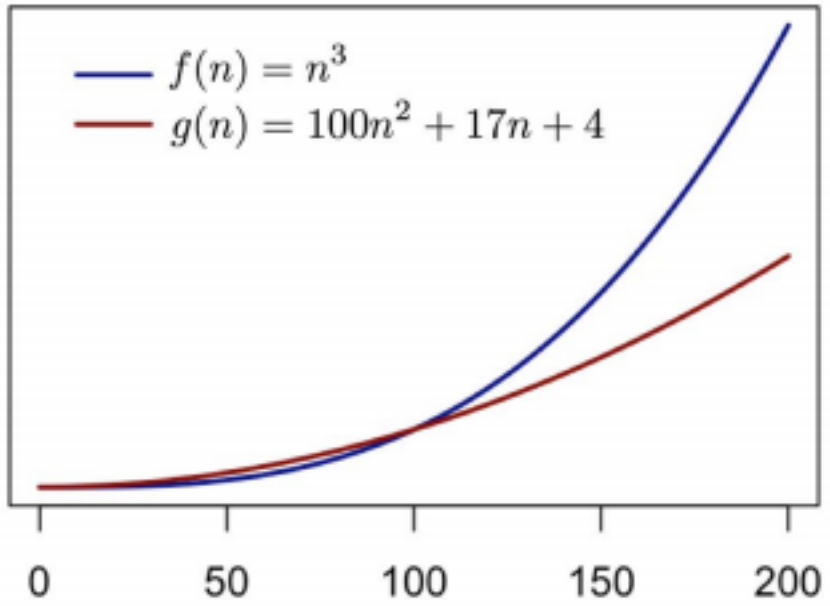
Algorithm Complexity – "big-O"

Definition: Let f and g be functions from the set of integers. We say that $f(n)$ is $O(g(n))$ if there are constants C and k such that

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

whenever $n > k$.

$g(n) \leq f(n)$ for $n > 100$



Algorithm Complexity – "big-O"

Example: Show that $100n^2 + 17n + 4$ is $O(n^2)$.

$100n^2 \leq 100n^2$ for all n

$17n \leq 17n^2$ for $n \geq 1$

$4 \leq 4n^2$ for $n \geq 1$

$100n^2 + 17n + 4 \leq 100n^2 + 17n^2 + 4n^2$

$= 121n^2$

$\Rightarrow 100n^2 + 17n + 4$ is $O(n^2)$ with $C = 121$ and $K = 1$

Alternate

$4 \leq n^2$ $n \geq 2$

$100n^2 + 17n + 4 \leq 118n^2$

$\Rightarrow O(n^2)$ with $C = 118$ and $K = 2$

Algorithm Complexity – "big-O"

- **Note:** $f(n)$ is $\mathcal{O}(g(n))$ means that for some C and k, $f(n)$ is bounded above by $C \cdot g(n)$
- This definition also means that $100n^2 + 17n + 4$ is $\mathcal{O}(n^3)$ and $\mathcal{O}(n^{1000})$...
- This is slightly unsatisfactory, because saying that $100n^2 + 17n + 4$ is $\mathcal{O}(n^{1000})$ is in the realm of information that is true, but not very useful.

Algorithm Complexity

upper bound

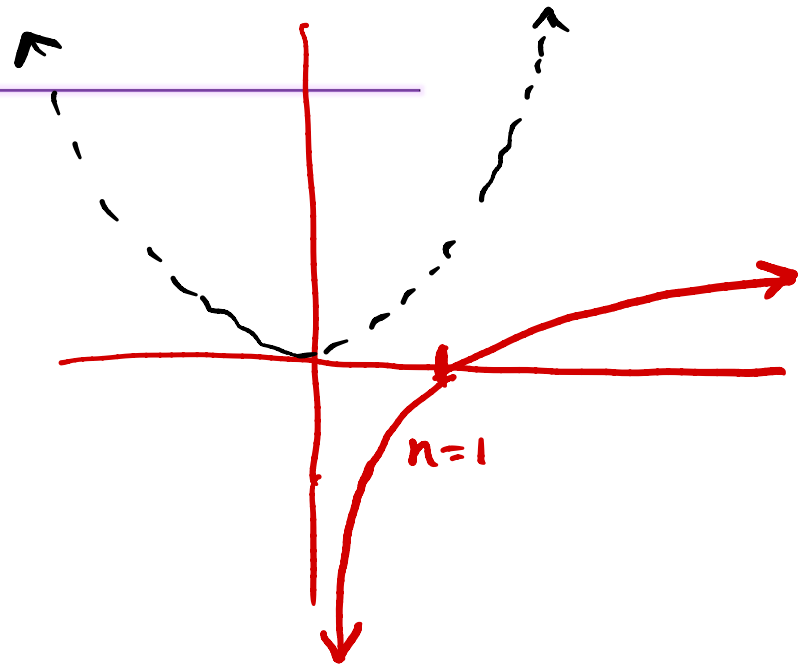
Example: Give the big-O estimate of $h(n) = 2n^2 + 5\log(n)$

$2n^2 \leq 2n^2$ for all n

$5\log n \leq 5n^2$ for all n

$h(n) \leq 2n^2 + 5n^2 = 7n^2$

$\Rightarrow$ $h(n)$ is $\theta(n^2)$ with $C = 7$ and $K = 1$



Algorithm Complexity – "big–O"

Theorem: Suppose that $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$.
Then $(f_1 + f_2)(n)$ is $O(\max(|g_1(n)|, |g_2(n)|))$.

Theorem: Suppose that $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$.
Then $(f_1 f_2)(n)$ is $O(g_1(n)g_2(n))$.

Algorithm Complexity – "big-Omega"

So $f(n)$ is $\mathcal{O}(g(n))$ tells us that f grows no faster than g

That's a nice upper bound on the growth of f

But it would be ***really*** nice to also have a **lower bound**… to say, for example, that $f(n)$ grows ***at least as fast as*** $h(n)$

Definition: Suppose that $f(n)$ and $h(n)$ are functions from the set of integers. We say that $f(n)$ is $\Omega\big(h(n)\big)$ if there are constants C and k such that

$$|f(n)| \geq C|h(n)|$$

whenever $n > k$.

Algorithm Complexity – "big-Omega" – lower bound

Example: Give a big-Ω estimate for $100n^2 + 17n + 4$.

$100n^2 \geq 100n^2$ for all n

$17n \geq 0$

$4 \geq 0$

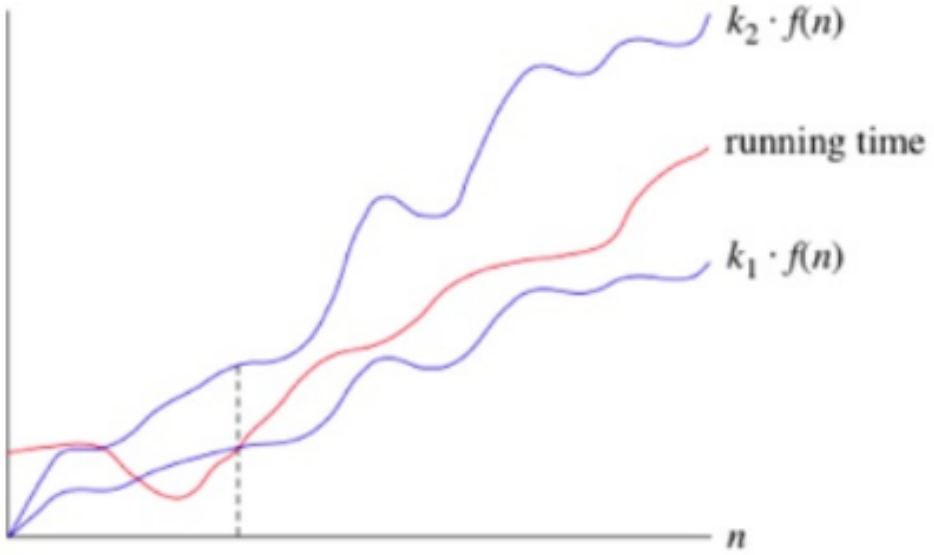
$100n^2 + 17n + 4 \geq 100n^2$ for $n > 0$

Therefore $100n^2 + 17n + 4$ is $\Omega(n^2)$ with $C = 100$ and $k = 1$.

Algorithm Complexity – "big–Theta"

Growth of Functions:

Def: If $f(n)$ is both $\mathcal{O}(g(n))$ and $\Omega(g(n))$ then we say $f(n)$ is $\Theta(g(n))$ [read big-Theta of $g(n)$] and also that $f(n)$ is of *order* $g(n)$



Algorithm Complexity – "big–Theta"

Big-Θ estimates of a function are the best of both worlds.

- They say we know everything about the growth of a function.
- It's smaller than "this", but larger than "that".

Definition: If $f(n)$ and $g(n)$ are both $\Theta(h(n))$, then we say that f and g have order $h(n)$, and that f and g are the same order.

Algorithm Complexity

Example: Show that $h(n) = 2n^2 + 5n \log n$ is $\Theta(n^2)$ $\quad n > \log n$ for $n \geq 1$

big-O: (upper bound)

$2n^2 \leq 2n^2$

$5n \log n \leq 5n \cdot n$ because $\log n < n$ for $n \geq 1$

$|h(n)| \leq 7n^2 \Rightarrow$ $h(n)$ is $O(n^2)$ with $c = 7, k = 1$

big-Ω (lower bound)

$2n^2 \geq 2n^2$

$5n \log n \geq 0$ for $n \geq 1$

$|h(n)| \geq 2n^2$

$\Rightarrow$ $h(n)$ is $\Omega(n^2)$ with $c = 2, k = 1$

Algorithm Complexity

Example (Continued) : Show that $h(n) = 2n^2 + 5n \log n$ is $\Theta(n^2)$

Since $h(n)$ is $O(n^2)$ and $\Omega(n^2)$

we can conclude that it is also $\Theta(n^2)$.

"order n^2"

Aside:

$\log(n) < n^{1/10}$ for $n > 10^{10}$

$\log_{10}(n) < n^{1/10}$

$\log_{10} 10^{10} = 10$ $\quad (10^{10})^{1/10} = 10$

Algorithm Complexity

Example: Give a big-O estimate of $f(n) = n!$

$$n! = n(n-1)(n-2)(n-3)(n-4)\cdots 5\cdot 4\cdot 3\cdot 2\cdot 1$$
$$\leq n\cdot n\cdot n\cdot n\cdot n\cdots n\cdot n\cdot n\cdot n\cdot n$$
$$\leq n^n$$

$\boxed{\Theta(n^n)}$ with $C=1$ and $k=1$

Example: Give a big-O estimate of $g(n) = \log n!$

$$\log(n!) \leq \log(n^n) \quad \text{for} \quad n \geq 1$$
$$= n\log(n)$$
$$\leq n^2 \quad \text{for} \quad n \geq 1 \Rightarrow \boxed{\Theta(n^2)} \text{ with } C=1 \text{ and } k=1$$

Matrices and Matrix Operations

Linear Algebra is the workhorse of computational science

In scientific and engineering code, 99% of computing time is spent on matrix operations

Although applications of matrices are incredibly broad, they were invented for a very simple purpose: to make solving systems of equations cleaner

$$\begin{aligned} 3x + 4y + 5z &= 1 \\ 2x + 8y + 3z &= 2 \\ 4x + 2y + 2z &= 3 \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Rectangular thing is a **matrix**, tall skinny things are **vectors**

Matrices and Matrix Operations

$$\begin{aligned} 3x + 4y + 5z &= 1 \\ 2x + 8y + 3z &= 2 \\ 4x + 2y + 2z &= 3 \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

A $\vec{x}$ b

Rectangular thing is a **matrix**, tall skinny things are **vectors**

Def: A matrix with m rows and n columns has **dimensions** $m \times n$

Rows x Columns

Def: A vector with n entries has **length** n

Notation: Matrices are represented by capital letters, like A and M.
Vectors are represented by lower case letters like $\mathbf{x}$ and $\mathbf{b}$

Example: The above **matrix equation** could be written as $A\mathbf{x} = \mathbf{b}$.

$A\vec{x} = \vec{b}$

Matrices and Matrix Operations

Matrices and vectors can be **added** and **multiplied** (but not divided)

Def: The sum of matrices A and B is the matrix obtained by adding corresponding entries together

Example:

A + B = C

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & -2 & 0 \\ 5 & 1 & -3 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 5 \\ 7 & 9 & 0 \\ 5 & 3 & 4 \end{bmatrix}$$

$c_{23} = 0$

$c_{33} = 4$

Note: Only makes sense if A and B have the **same** dimensions

Notation: We refer the the entry in the i^{th} row and j^{th} column of the matrix A as a_{ij} or $A[i][j]$

Matrices and Matrix Operations

Complexity of Matrix Addition:

Simple calculation. We add each pair of entries

There are $n \cdot n = n^2$ entries

So matrix addition requires n^2 additions (which is $\Theta(n^2)$)

Matrices and Matrix Operations

```
procedure MatrixAddition(A, B)
C := 0        # Initialize C to n × n zero matrix
for i := 1 to n
    for j := 1 to n
        C[i][j] := A[i][j] + B[i][j]
return C
```

Turn loop into summations and count number of basic operations

Complexity: $\sum_{i=1}^{n} \sum_{j=1}^{n} 1 = \sum_{i=1}^{n} n = n^2$

Matrices and Matrix Operations

$$\begin{aligned} 3x + 4y + 5z &= 1 \\ 2x + 8y + 3z &= 2 \\ 4x + 2y + 2z &= 3 \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Matrices can **multiply** vectors, resulting in a new vector.

Operation is defined directly from the analogy between matrix equation and system of equations

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \Leftrightarrow \quad \begin{aligned} &3x + 4y + 5z \\ &2x + 8y + 3z \\ &4x + 2y + 2z \end{aligned}$$

Matrices and Matrix Operations

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 4 \cdot 2 + 5 \cdot 3 \\ 2 \cdot 1 + 8 \cdot 2 + 3 \cdot 3 \\ 4 \cdot 1 + 2 \cdot 2 + 2 \cdot 3 \end{bmatrix} = \begin{bmatrix} 26 \\ 27 \\ 14 \end{bmatrix}$$

Think of it as putting the vector over the matrix, multiplying down the columns, and adding

$$\begin{matrix} 1 & 2 & 3 \\ \downarrow & \downarrow & \downarrow \end{matrix}$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 4 \cdot 2 + 5 \cdot 3 \\ 2 \cdot 1 + 8 \cdot 2 + 3 \cdot 3 \\ 4 \cdot 1 + 2 \cdot 2 + 2 \cdot 3 \end{bmatrix}$$

Note: Length of vector **must equal** num of columns in matrix

Matrices and Matrix Operations

Example: Compute $A\mathbf{x}$ where

$$A = \begin{bmatrix} 1 & -2 \\ 2 & 3 \\ -1 & 5 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$A\vec{x} = \begin{bmatrix} 1 & -2 \\ 2 & 3 \\ -1 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + -2 \cdot 3 \\ 2 \cdot 1 + 3 \cdot 3 \\ -1 \cdot 1 + 5 \cdot 3 \end{bmatrix} = \begin{bmatrix} -5 \\ 11 \\ 14 \end{bmatrix}$$

Matrices and Matrix Operations

Pseudocode and Complexity: Let A be $n \times n$ and $\mathbf{x}$ be length n

procedure `MatVec`(A, x)

$y := \mathbf{0}$ # Initialize y to n-length zero vector

for $i := 1$ to n # Do one row at a time

  for $j := 1$ to n # Loop over entries in i^{th} row

    $y[i] \; += \; A[i][j] * x[j]$ # Multiply and accumulate in y

return y

Matrices and Matrix Operations

procedure $\texttt{MatVec}(A, x)$

$y := \mathbf{0}$ # Initialize y to n-length zero vector

for $i := 1$ to n # Do one row at a time

  for $j := 1$ to n # Loop over entries in i^{th} row

    $y[i] \; += \; A[i][j] * x[j]$ # Multiply and accumulate in y

return y

Could count additions and multiplications. Usually combine and count FLOPs (floating point operations) instead

Complexity: $\sum_{i=1}^{n} \sum_{j=1}^{n} 2 = \sum_{i=1}^{n} 2n = 2n^2$

Matrices and Matrix Operations

We can also multiply matrices together

Example:

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ 3 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 14 & 14 & 16 \\ 25 & 14 & 21 \\ 4 & 6 & 10 \end{bmatrix}$$

Process: Think of it as multiple matrix-vector products

$$AB = A\,[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3\,] = [A\mathbf{b}_1 \quad A\mathbf{b}_2 \quad A\mathbf{b}_3\,]$$

Question: Which dims of A and B must match for this to work?

Matrices and Matrix Operations

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix}\begin{bmatrix} \mathbf{-1} & 0 & 1 \\ \mathbf{3} & 1 & 2 \\ \mathbf{1} & 2 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{14} & 14 & 16 \\ \mathbf{25} & 14 & 21 \\ \mathbf{4} & 6 & 10 \end{bmatrix}$$

Process: Think of it as multiple matrix-vector products

$$AB = A\,[\mathbf{b}_1 \;\; \mathbf{b}_2 \;\; \mathbf{b}_3] = [A\mathbf{b}_1 \;\; A\mathbf{b}_2 \;\; A\mathbf{b}_3]$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix} =$$

Matrices and Matrix Operations

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} -1 & \mathbf{0} & 1 \\ 3 & \mathbf{1} & 2 \\ 1 & \mathbf{2} & 1 \end{bmatrix} = \begin{bmatrix} 14 & \mathbf{14} & 16 \\ 25 & \mathbf{14} & 21 \\ 4 & \mathbf{6} & 10 \end{bmatrix}$$

Process: Think of it as multiple matrix-vector products

$$AB = A\,[\mathbf{b}_1 \;\; \mathbf{b}_2 \;\; \mathbf{b}_3] = [A\mathbf{b}_1 \;\; A\mathbf{b}_2 \;\; A\mathbf{b}_3]$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} =$$

Matrices and Matrix Operations

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 & \mathbf{1} \\ 3 & 1 & \mathbf{2} \\ 1 & 2 & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 14 & 14 & \mathbf{16} \\ 25 & 14 & \mathbf{21} \\ 4 & 6 & \mathbf{10} \end{bmatrix}$$

Process: Think of it as multiple matrix-vector products

$$AB = A\,[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3\,] = [A\mathbf{b}_1 \quad A\mathbf{b}_2 \quad A\mathbf{b}_3\,]$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} =$$

Matrices and Matrix Operations

$$AB = A\,[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3\,] = [A\mathbf{b}_1 \quad A\mathbf{b}_2 \quad A\mathbf{b}_3\,]$$

Question: Which dims of A and B must match for this to work?

Question: What are the dimensions of $C = AB$?

Summary: If A is $m \times n$ and B is $n \times k$ then AB is $m \times k$

Matrices and Matrix Operations

Example: Find the **Complexity**: $C = AB$ where A and B are both $n \times n$

Matrices and Matrix Operations

Summary:

- Matrix addition is $\Theta(n^2)$
- Matrix-vector multiplication is $\Theta(n^2)$
- Matrix-Matrix multiplication is $\Theta(n^3)$

What we've done:

- ❖ Complexity of Algorithms
- ❖ Matrix Operations

Next:

Induction!

What we've done:

- ❖ Complexity of Algorithms
- ❖ Matrix Operations

Next:

Induction!