## Announcements

- Midterm will be a take-home exam, due Monday beginning of class.

## More Complexity

Quick recap:

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

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

whenever $n > k$. ("$f(n)$ is big-O of $g(n)$")

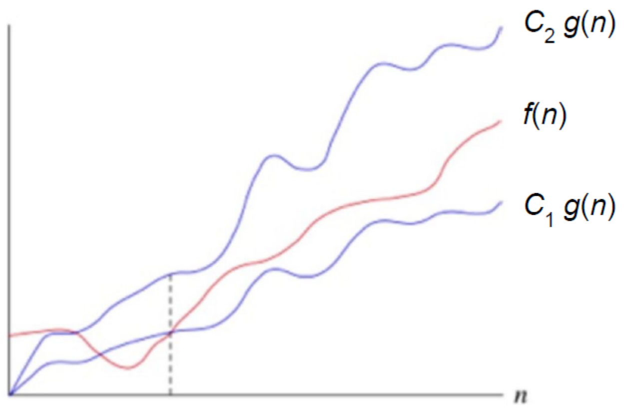
**Definition** Suppose that $f(n)$ and $h(n)$ are functions from the set of integers. We say that $f(n)$ is $\Omega(h(n))$ if there are constants $C$ and $k$ such that

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

whenever $n > k$. ("$f(n)$ is big-Omega of $h(n)$")

**Definition** If $f(n)$ is both $\mathcal{O}(g(n))$ and $\Omega(g(n))$ then we say $f(n)$ is $\Theta(g(n))$, and also say that $f(n)$ is of **order** $g(n)$.

The figure below is an example of a function $f(n)$ that is $\Theta(g(n))$




Example: Show that $h(n) = 8n^2 - 2n\log n$ is $\Theta(n^2)$

First we show that $h(n)$ is $\mathcal{O}(n^2)$, then show that $h(n)$ is $\Omega(n^2)$.

*Proof.* Showing $h(n)$ is $\mathcal{O}(n^2)$...

For $n > 1$ we know that $n \cdot \log n > 0$, which implies that subtracting this term is only making $h$ smaller. So we have

$$h(n) = 8n^2 - 2n\log n \leq 8n^2$$

$\Rightarrow h$ is $\mathcal{O}(n^2)$ with witnesses $C = 8$ and $k = 1$

Showing $h(n)$ is $\Omega(n^2)$...

We know that for $n > 1$, $\log n < n$. Then $n\log n < n^2$ and $-2n\log n > -2n^2$. This implies that

$$h(n) > 8n^2 - 2n^2 = 6n^2$$

$\Rightarrow h$ is $\Omega(n^2)$ with $C = 6$ and $k = 1$.

$\therefore h(n)$ is $\Theta(n^2)$ $\square$

**Matrices and matrix operations**

- Linear algebra is the workhorse of computational science
- In scientific computing, a huge amount of computing time is spent on matrix operations.
- Applications of matrices are broad, but they were invented for a simple purpose: to make solving systems of equations cleaner.

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

- Rectangular thing is a **matrix**, tall skinny things are **vectors**
- **Definition** A matrix with $m$ rows and $n$ columns has **dimensions** $m \times n$
- **Definition** A vector with $n$ entries has **length** $n$
- **Notation** Matrices are represented by capital letters, like $A$ and $M$. Vectors are represented by lowercase letters like $\mathbf{x}$ and $\mathbf{b}$ (often bold-faced)
- Example: The above **matrix equation** could be written as $A\mathbf{x} = \mathbf{b}$

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

**Definition** The **sum** of matrices $A$ and $B$ is the matrix obtained by adding the corresponding entries of each matrix together

Example:

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 8 & 3 \\ 4 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 4 & 6 & 8 \\ 6 & 13 & 9 \\ 11 & 10 & 11 \end{bmatrix}$$

Note: This is only defined if $A$ and $B$ have the *same* dimensions.

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

**Complexity of matrix addition:**

- Straightforward: We add each pair of entries.

  $\Rightarrow$ for two $m \times n$ matrices, there are $mn$ entries, so $mn$ additions

  $\Rightarrow$ for square matrices of size $n \times n$, thats $n^2$ additions, so this is $\mathcal{O}(n^2)$.

```
def matrixAdd(A, B):
    S = 0 # S is the 0 matrix with same dimension as A and B
    for i in 1,num_rows:
        for j in 1,num_cols:
            S[i,j] = A[i,j]+B[i,j]
    return (S)
```

We see that for each iterate in the inner loop we have one addition. So adding up all the iterations, we get

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

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

- Think of it as taking the vector and setting it down on top of the matrix.

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

Each member of the vector is multiplied by its corresponding column of the matrix and the resulting columns are added together.

**Rule:** This means that the length of the vector *must equal* the number of columns of the matrix

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

$$A = \begin{bmatrix} 1 & 3 \\ -2 & 4 \\ 1 & 5 \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Multiplying the elements of $\mathbf{x}$ over the corresponding columns of $A$ and adding the resulting columns we get

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

**Complexity:** Intuition and estimation – counting multiplications and additions, what is a rough estimate of the complexity?

- Given an $n \times m$ matrix, for each row, we seem to have $n$ multiplications and $n-1$ additions. ($2n-1$ total operations).
- Times $m$ rows, we get $2nm - m$ total operations
- Or if it is a square $n \times n$ matrix, $2n^2 - n$, or we might say $\mathcal{O}(n^2)$

Let $A$ be $n \times n$ and let $\mathbf{x}$ be length $n$.

```
def matrixAdd(A, x, n):
    y = 0 # y is a 0 vector of length n
    for i in 1,n:
        y[i] = A[i][1]*x[1]
        for j in 2,n:
            y[i] += A[i][j]*x[j]

    return (y)
```

Let's count the additions and multiplications. (Usually we conflate these for the sake of analyzing complexity and call them FLOPs–Floating Point Operations)

We see that for each iterate in the outer loop, we have one multiplication and for each iterate in the inner loop we have one addition and one multiplication (two FLOPs). So the complexity is given by

$$\sum_{i=1}^{n}\left(1+\sum_{j=1}^{n-1}2\right) = \sum_{i=1}^{n}2(n-1)+1 = n(2n-1) = 2n^2-n$$

**Multiplying matrices:** Think of this as doing a few matrix-vector multiplications.

$$AB = \begin{bmatrix} 1 & 3 & -1 \\ -2 & 4 & 0 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 3 & -1 & 2 \\ 1 & 0 & -2 \end{bmatrix} = ?$$

$$AB = A\begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix} = \begin{bmatrix}A\mathbf{b}_1 & A\mathbf{b}_2 & A\mathbf{b}_3\end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & -1 \\ -2 & 4 & 0 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 3 & -1 & 2 \\ 1 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 10 & -2 & 8 \\ 8 & -6 & 8 \\ 19 & -4 & 6 \end{bmatrix}$$

**Question:** What must be the dimensions of $A$ and $B$ for this to work?

**Answer:** $A: \ m \times \mathbf{n} \Rightarrow B: \ \mathbf{n} \times k$ (The number of columns of $A$ must match the number of rows of $B$).

**Question:** So then what are the dimensions of $C = AB$ ($A$ is $m \times n$ and $B$ is $n \times k$)

**Answer:** $C$ must be $m \times k$

**Complexity:** If we multiply two $n \times n$ square matrices $A$ and $B$, then

$$AB = A\begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix} = \begin{bmatrix}A\mathbf{b}_1 & A\mathbf{b}_2 & A\mathbf{b}_3\end{bmatrix}$$

- Each column of $AB$ is a “mat-vec”
- We saw that each requires $2n^2 - n$ FLOPs
- And we have $n$ columns to do

  $\Rightarrow$ Total mat-mat is $n \cdot (2n^2 - n) = 2n^3 - n^2$ FLOPs

**Summary:**

- Matrix addition is $\mathcal{O}(n^2)$
- Matrix-vector multiplication (mat-vec) is $\mathcal{O}(n^2)$
- Matrix-matrix multiplication (mat-mat) is $\mathcal{O}(n^3)$