UNIVERSITY OF CALGARY

CPSC 319
Data Structures, Algorithms, and Their Applications

Winter 2024

Recursion

- A programming concept where a function calls itself to solve a problem.
- Continues until a base case is reached, returning a result without further recursive calls.

- Factorial:

  Definition:

  $n! = n \times (n-1) \times \ldots \times 2 \times 1$

  Recursive relation:

  $f(n) = f(n-1) \times n$

  $f(0) = 1$

Matrix

- A matrix is a two-dimensional array of numbers, symbols, or expressions arranged in rows and columns.
- Denoted by an uppercase letter, such as A.
- Matrix Elements: a_{ij} represents the element in the i-th row and j-th column of matrix A.
- Matrix Size: m×n indicates a matrix with m rows and n columns.

$$\mathbf{A} = \begin{bmatrix} -1.3 & 0.6 \\ 20.4 & 5.5 \\ 9.7 & -6.2 \end{bmatrix}.$$

Matrix Operations

- Addition:
  C = A + B for matrices A, B, and C of the same size.
  $c_{ij} = a_{ij} + b_{ij}$
- Subtraction:
  C = A - B for matrices A, B, and C of the same size.
  $c_{ij} = a_{ij} - b_{ij}$
- Scalar Multiplication:
  C = k · A for a scalar k and matrix A.
  $c_{ij} = k \cdot a_{ij}$
- Matrix Multiplication:
  C = A × B for matrices A, B, and C with appropriate dimensions.
  $c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$
- Transpose:
  $B = A^T$ where B is the transpose of A.
  $b_{ij} = a_{ji}$

Exponentiation

- Exponentiation is the process of raising a number (base) to a certain power (exponent).
- Definition:

  $n^m = n \times n \times \ldots \times n$ (m times)

- Recursive relation 1:

  $f(n, m) = f(n, m-1) \times n$

  $f(n, 1) = n$
- Recursive relation 2:

  If m is odd:

  $f(n, m) = f(n, m/2) \times f(n, m/2) \times n$

  If m is even:

  $f(n, m) = f(n, m/2) \times f(n, m/2)$

  $f(n, 1) = n$

System.nanoTime()

- Precision:
  - Provides nanosecond precision.
  - Suitable for measuring short durations or intervals.
- Origin:
  - The origin (zero point) is arbitrary and may change between runs or system restarts.
  - Not tied to the system clock; it measures elapsed time from an unspecified point in the past.
- Use Cases:
  - Measuring short intervals or time spans.
  - Profiling and performance monitoring.
  - Benchmarking code execution time.

System.currentTimeMillis()

- Precision:
  - Provides millisecond precision.
  - Suitable for measuring longer durations.
- Origin:
  - The origin is the standard Unix epoch (January 1, 1970, UTC).
  - Tied to the system clock and reflects wall-clock time.
- Use Cases:
  - Dealing with real-world time, such as timestamps.
  - Calculating date differences.
  - Scheduling tasks based on absolute time.