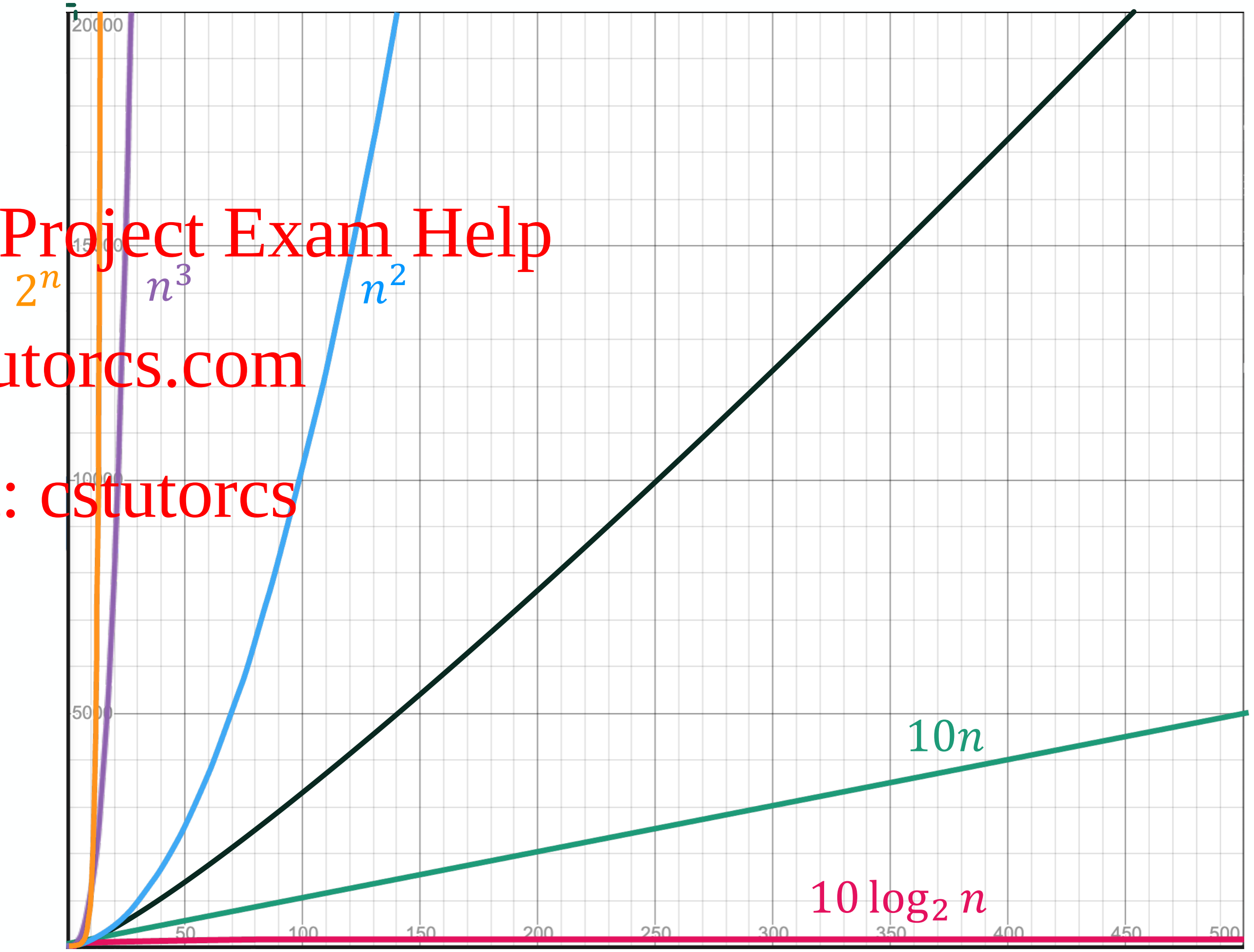
CIT 596: ALGORITHMS & COMPUTATION

Comparing Common Running Times

Common Running Times

Certain asymptotic growth rates show up very often in the analysis of algorithms.

- Constant: $O(1)$
- Logarithmic: $O(\log n)$
- Linear: $O(n)$
- Linearithmic: $O(n \log n)$
- Quadratic: $O(n^2)$
- Polynomial: $O(n^k)$
- Exponential: $O(2^n)$



Constant: $O(1)$

- **Running time of:** Basic operations like arithmetic with bounded-length numbers, accessing an array entry, or assigning a value to a variable.
- **Example of a $\Theta(1)$ function:** $f(n) = 2$
  - $f(1000) = 2$
- **Note:**
  - $c = O(g(n))$ for every constant c and non-decreasing function g.

Logarithmic: $O(\log n)$

- **Running time of:**
  - binary search on a sorted array
  - heap operations

- **Example of a $\Theta(\log n)$ function:** $f(n) = 2\log_2(n) + 4$
  - $f(1000) \approx 24$

- **Notes:**
  - $\log_a(n) = O(n^\epsilon)$ for all $a > 1$ and $\epsilon > 0$.
  - $\log_a(n) = \Theta(\log_b(n))$ for all $a, b > 1$.

Linear: $O(n)$

- **Running time of:**
  - Addition of n-digit numbers
  - Huffman's algorithm for data compression
- **Example of a $\Theta(n)$ function:** $f(n) = 2n + 1$
  - $f(1000) = 2001$
- **Note:**
  - It takes $\Theta(n)$ time just to read the whole input.

Linearithmic: $O(n \log n)$

- **Running time of:**
  - Merge sort
  - Expected running time of randomized quicksort
- **Example of a $\Theta(n \log n)$ function:** $f(n) = 2n \log_3 n$
  - $f(1000) \approx 12575$
- **Note:**
  - Close to linear and still considered quite fast for most applications.

Quadratic: $O(n^2)$

- **Running time of:**
  - Insertion sort
  - DNA sequence alignment algorithms
  - Many algorithms with nested loops
- **Example of a $\Theta(n^2)$ function:** $f(n) = n^2 + 3n + 1$
  - $f(1000) = 1003001$

Polynomial: $O(n^k)$

for any positive integer k

- **Running time of:**
  - Finding shortest paths between all pairs of nodes in a graph ($O(n^3)$)
  - Many algorithms with triple-nested loops ($O(n^3)$)
  - Many algorithms with quadruple-nested loops ($O(n^4)$), etc.
- **Example of a $\Theta(n^4)$ function:** $f(n) = n^4 + 2n^3 + n$
  - $f(1000) = 1002000001000$
- **Notes:**
  - $n^k = O(n^\ell)$ if $k \le \ell$.
  - If a computational problem has a polynomial-time algorithm, we generally consider the problem **tractable**.

Exponential: $O(2^n)$

- **Running time of:**
  - Towers of Hanoi with n rings
  - Finding all subsets of a set of size n
- **Example of a $\Theta(2^n)$ function:** $f(n) = 2^n$
  - $f(1000) =$10715086071862673209484250490600018105614048117055336074437503883703510511249361224931983788156958581275946729175531468251871452856923140435984577574698574803934567774824230985421074605062371141877954182153046474983581941267398767559165543946077062914571196477686542167660429831652624386837205668069376
- **Notes:**
  - $2^n = \Omega(n^k)$ for all k.
  - Running times like $O(n!)$, $O(n^n)$, or $O\left(2^{n^2}\right)$ are even more disastrously slow.