COT 6405
ANLYSIS OF ALGORITHMS

A Survey of Common Running Times

Computer & Electrical Engineering and Computer Science Dept.
Florida Atlantic University

Spring 2017

Common order of growth functions

Theta Form	*Name*
$\Theta(1)$	Constant
$\Theta(\lg \lg n)$	Log log
$\Theta(\lg n)$	Log
$\Theta(n^c),\ 0 < c < 1$	Sublinear
$\Theta(n)$	Linear
$\Theta(n \lg n)$	$n \log n$
$\Theta(n^2)$	Quadratic
$\Theta(n^3)$	Cubic
$\Theta(n^k),\ k \geq 1$	Polynomial
$\Theta(c^n),\ c > 1$	Exponential
$\Theta(n!)$	Factorial

Sublinear Time

- RT is asymptotically smaller than linear
- Reading the input takes linear time
- Occurs when:
  - Input “queried” indirectly rather than read completely
  - Try to minimize the number of queries
- Example: *Given a sorted array A of n numbers, determine whether a given number p belongs to the array.*
  - traverse the array $\Rightarrow \Theta(n)$
  - the *binary search algorithm* takes $\Theta(\log n)$

Linear Time $\Theta(n)$

- One-pass or a constant number of passes through the input elements
- Examples:
  - Find the max/min of *n* numbers
  - Merge two sorted arrays into one sorted array

O(nlogn) Time

- a very common RT
- any algorithm that splits its input into two equal-sized pieces, solve each piece recursively, then combine the two solutions in linear time
  - Example: Merge-Sort algorithm

    $T(n) = 2 \cdot T(n/2) + \Theta(n)$

    $T(n) = \Theta(n\log n)$

Quadratic Time, $\Theta(n^2)$

- Example problem: *given n points in the plane, find the closest pair of points*

```
for each input point (x_i, y_i)
    for each other input point (x_j, y_j)
        compute distance d = sqrt((x_i - x_j)^2 + (y_i - y_j)^2)
        if d is < than the current min, then min = d
return min
```

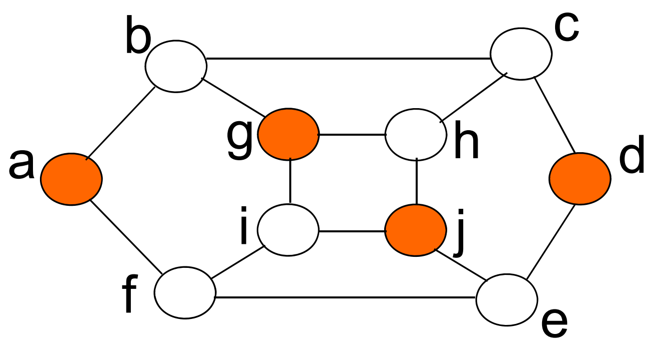
$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

RT = $\Theta(n^2)$

Polynomial Time, $O(n^k)$ where k - constant

- Example problem: *Given a graph G with n nodes, find whether G has an **independent set** of size k.*

A set S of nodes in G, $S \subseteq V$, is <u>independent</u> if no two nodes in S are joined by an edge.



S = {a, g, j, d} is an independent set of size 4

Polynomial Time, $O(n^k)$ where k - constant

- take all the groups of k nodes and check if any group forms an independent set
- the number of groups of k nodes is $\binom{n}{k} = \Theta(n^k)$
- to check if a set of k nodes forms an independent set takes $\binom{k}{2} = \Theta(k^2)$

- since k is constant, the overall RT = $\Theta(n^k)$

Exponential Time, $\Theta(c^n)$ where c - constant

- Example problem: *Given a graph G with n nodes, find an independent set of maximum size*

- take all subsets of nodes and return the one with maximum size which is an independent set
- the number of subsets is 2^n
- to check if a set of n nodes is independent takes $\binom{n}{2} = \Theta(n^2)$

- The overall RT = $\Theta(2^n n^2)$

Factorial Time, $\Theta(n!)$

- Example: Traveling Salesman Problem

Given a set of n cities, with distances between all pairs, what is the shortest tour that visits all cities?

- NP – complete problem
- enumerate all possible tours, then chose the shortest one
- RT = $\Theta(n!)$