Evaluation

• Exams (2)	50%
• Homework Assignments	36%
• Semester Project	13%
• Class Participation	1%

Celebrity Problem

- A **Celebrity** is one that knows nobody and that everybody knows.

Celebrity Problem:

INPUT**:** n persons with a $n \times n$ information matrix.

OUTPUT**:** Find the "celebrity", if one exists.

MODEL**:** Only allowable questions are:

  – ***Does person i know person j?***

- Naive Algorithm: O(n^2) Questions.

Celebrity Problem (Cont'd)

- Naive Algorithm: $O(n^2)$ Questions.
  - Ask everyone of everyone else for a total of n(n-1) questions
- Using Divide-and-Conquer: $O(n \log_2 n)$ Questions.
  - Divide the people into two equal sets. Solve recursively and find two candidate celebrities from the two halves. Then verify which one (if any) is a celebrity by asking n-1 questions to each of them and n-1 questions to everyone else about them. This gives a recurrence for the total number of questions asked: $T(n) = 2T(n/2) + 2n$
- Improved solution?
  - How many questions are needed to find a non-celebrity?
  - Hint: What information do you gain by asking one question?

Celebrity Problem (Cont'd)

– **Induction Hypothesis 2:** We know how to find n-2 non-celebrities among a set of n-1 people, i.e., we know how to find at most one person among a set of n-1 people that could potentially be a celebrity.

– Resulting algorithm needs [3(n-1)-1] questions.

Definitions

Algorithm: It is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. It is the sequence of computational steps that transform the input into the output.

Instance of a Problem: consists of the input (satisfying whatever constraints are imposed in the problem statement) needed to compute a solution to the problem.

Correct Algorithm: for every input instance it halts with the correct output. A correct algorithm solves the given computational problem.

Solving Recurrence Relations

Page 62, [CLR]

Recurrence; Cond	Solution
$T(n) = T(n-1) + O(1)$	$T(n) = O(n)$
$T(n) = T(n-1) + O(n)$	$T(n) = O(n^2)$
$T(n) = T(n-c) + O(1)$	$T(n) = O(n)$
$T(n) = T(n-c) + O(n)$	$T(n) = O(n^2)$
$T(n) = 2T(n/2) + O(n)$	$T(n) = O(n \log n)$
$T(n) = aT(n/b) + O(n)$; $a = b$	$T(n) = O(n \log n)$
$T(n) = aT(n/b) + O(n)$; $a < b$	$T(n) = O(n)$
$T(n) = aT(n/b) + f(n)$; $f(n) = O(n^{\log_b a - \epsilon})$	$T(n) = O(n)$
$T(n) = aT(n/b) + f(n)$; $f(n) = O(n^{\log_b a})$	$T(n) = \Theta(n^{\log_b a} \log n)$
$T(n) = aT(n/b) + f(n)$; $f(n) = \Theta(f(n))$ $af(n/b) \leq cf(n)$	$T(n) = \Omega(n^{\log_b a} \log n)$