COT 5407: Introduction to Algorithms

Giri Narasimhan

ECS 389; Phone: x3748

giri@cs.fiu.edu

www.cs.fiu.edu/~giri/teach/5407F05.html

Quote by Charles Babbage

As soon as an *Analytical Engine* exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will then arise — by what course of calculation can these results be arrived at by the machine in the shortest time?

Charles Babbage (1864)

History of Babbage:
http://ei.cs.vt.edu/~history/Babbage.html

Evaluation

• Exams (2)	50%
• Homework Assignments	35%
• Semester Project	10%
• Class Participation	5%

Search

- You are asked to guess a number X that is known to be an integer lying between integers A and B. How many guesses do you need in the worst case?
  - Use binary search; Number of guesses = $\log_2(B-A)$
- You are asked to guess a positive integer X. How many guesses do you need in the worst case?
  - NOTE: No upper bound is known for the number.
  - Algorithm:
    - figure out B (by using Doubling Search)
    - perform binary search in the range B/2 through B.
  - Number of guesses = $\log_2 B + \log_2(B - B/2)$
  - Since X is between B/2 and B, we have: $\log_2(B/2) < \log_2 X$,
  - Number of guesses $< 2\log_2 X - 1$

Polynomials

- Given a polynomial
  - $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n$

  compute the value of the polynomial for a given value of x.
- How many additions and multiplications are needed?
  - Simple solution:
    - Number of additions = n
    - Number of multiplications = 1 + 2 + … + n = n(n+1)/2
  - Improved solution using **Horner's rule**:
    - $p(x) = a_0 + x(a_1 + x(a_2 + \ldots x(a_{n-1} + x\, a_n))\ldots))$
    - Number of additions = n
    - Number of multiplications = n

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.
- Using Divide-and-Conquer: $O(n \log_2 n)$ Questions.
- Improved solution?

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: $T(n) = 2T(n/2) + 4(n-1)$
- Improved solution?
  - 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.