Introduction

Design and Analysis of Algorithms
Andrei Bulatov

Course Info

- **Instructor: Andrei Bulatov**
  - Email: abulatov@cs.sfu.ca
  - Room: TASC 8013
  - Office hours (tentative):
    Thursday 3:30 – 17:00
- **Teaching Assistant:**
  - Carrie Wang, email: cwa9@sfu.ca
- **Course webpage**
  - http://www.cs.sfu.ca/CC/405/abulatov

Course Info

- **Course objective:**
  To introduce more advanced algorithmic techniques, methods of algorithm analysis, and models of computation
- **Syllabus:**
  - Review of models of computation, dynamic programming, greedy algorithms
  - Graph algorithms and network flow
  - Branch and bound
  - NP-Completeness
  - Approximation algorithms
  - Randomized algorithms
  - Algorithmic game theory, Markov chains, Monte Carlo method, fast Fourier transform (if time permits)

Course Info

- **Textbook:**
  - Cormen, Leiserson, Rivest, Stein,
    Introduction to Algorithms, McGraw Hill, MIT Press.
  - Kleinberg, Tardos,
    Algorithm Design, Addison Wesley
  - It is impossible to finish studying all the contents of the textbook in one semester. The contents not covered in lectures/slides are not required, unless explicitly indicated as required.
  - The content and order of topics, as presented in the class, do not one-to-one correspond to any part of the books. Use of Subject Index and Recommended Text is advised.

Course Info

- **References:**
  - D. E. Knuth, *The Art of Computer Programming. Vol. 1,2,3,4,* Addison-Wesley
  - R. L. Graham; D. E. Knuth; and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, Reading, MA, 1994

Course Info

- **Grading:**
  - 8 Assignments ($8 \times 3\%$)
  - 1 Midterm 26%
  - 1 Final Exam 50%

Prerequisites

- Basic knowledge of algorithms
- Some general knowledge is needed, as there will be examples
- Basic math erudition
- Some experience in programming is very helpful

Closest Pair: The Problem

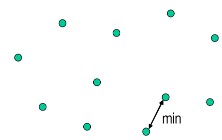
The Closest Pair Problem

Instance:

n points in the plane

Objective:

Find a pair of points that are closest together



Closest Pair: Algorithm

Input:

points $a_1,\dots,a_n$ in the plane

Output:

pair a_i, a_j such that $|a_i a_j|$ is minimal

Method:

set $a = a_1$, $b = a_2$ and $d = |a_1 a_2|$
for $1 \le i < j \le n$ do
  if $|a_i a_j| < d$ then set $a = a_i$, $b = a_j$ and $d = |a_i a_j|$
endfor
output a,b

Closest Pair: Soundness

An algorithm is sound / correct if it outputs what it is supposed to output

Theorem

The Closest Pair algorithm is sound.

(In other words for any input points it returns a pair of points that are at minimal distance between them.)

Proof.

trivial

Closest Pair: Running Time

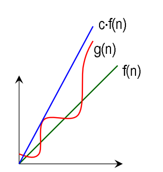
Evaluating running time is not easy

In the Closest Pair algorithm the **for** loop is executed $\frac{n(n-1)}{2}$ times

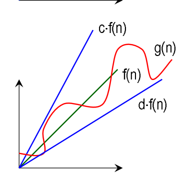
However each execution requires several elementary steps, so we tend to say that its running time is somewhat like n^2

Asymptotic Notation

- For two functions f,g: $\mathbb{N} \to \mathbb{R}$



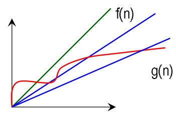
- g is in O(f) if there is c such that starting from some k: g(n) ≤ c · f(n)



- g is in Θ(f) if there are c,d such that starting from some k: d · f(n) ≤ g(n) ≤ c · f(n)

Asymptotic Notation

- g is in o(f) if for any c starting from some k(c): g(n) < c · f(n)



- The running time of Closest Pair is O(n^2)
- Other frequent running times:
  - linear O(n)
  - O(n log n)
  - polynomial O(n^k)
  - exponential O(2^{cn})
  - sublinear o(n)

Closest Pair: Real Running Time

Do you see anything wrong with our O(n^2) estimation?