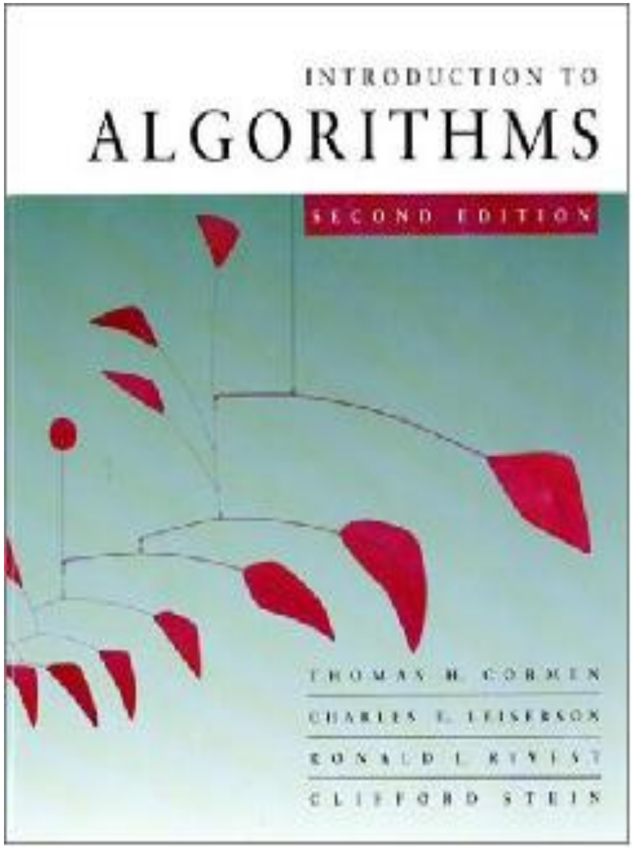
CIS507 DESIGN & ANALYSIS OF ALGORITHMS

Khaled Elbassioni
kelbassioni@masdar.ac.ae

Material

- Main Textbook:
  - Cormen, Leiserson, Rivest, Stein, ***Introduction to Algorithms***. *MIT Press, 3rd Edition, 2009*

- Classroom is to present ideas + discussion
- Detailed study is your responsibility



Assessment

No make-up quizzes/exams

No extensions

Final is Inclusive!

Item	Weight	Date	Due?
Quiz 1	5%	Jan 16	-
Quiz 2	5%	3 Feb	-
Quiz 3	5%	17 Feb	-
Homework 1	10%	17 Feb	1 March
Quiz 4	5%	6 Mar	-
Homework 2	10%	6 March	22 Mar
Mid-Term Exam	20%	24 Mar	-
Quiz 5	5%	24 April	-
Homework 3	10%	24 April	5 May
Final	30%	11-15 May	-

Getting Help

- Office hours:
  - 14:00-15:00 on class days
- Outside office hours:
  - By appointment
- Teaching Assistants
  - Mukesh Jha: mjha@masdar.ac.ae
  - Hayk Baluyan: bhayk@masdar.ac.ae

- Please feel free to interrupt me at anytime to ask questions!

Week	Topic (tentative outline)	Chapter
1	Intro, asymptotic notation, recurrences, master method	1, 2, 3,4
1	Divide and conquer, randomized sort	4, 5
2	Linear-time sorting, order statistics	6, 7, 8, 9
4	Hashing	11
5	Binary search trees, Red-Black Trees	12, 13
6	Augmenting data structures	14
7,8	Amortized analysis, competitive analysis	17
9	Dynamic programming, greedy algorithms	15, 16, 23
10,11	Shortest paths	22, 24, 25
12	Mid Term Exam	
15	Network flows, linear programming	26
16	NP Completeness, Reductions	34
17	Approximation algorithms	35
18	Revision & pre-exam break	
19	Final Exam	

Standard Question: *What's in the Exam?*

- **All** topics we cover are included in the exam
- Use your own judgment to determine what I consider most important:
  - Techniques & ideas covered in class are most fundamental

    they will constitute majority of questions
  - But the book is your ultimate source of material
  - In other words: Topics not covered in class, but are in the book

    included, but will constitute small proportion of grade
- I will indicate which chapters are covered, and which sections are excluded

Pre-Requisites

- Math
  - Set theory and discrete mathematics
  - Writing proofs (including induction and proof-by-contradiction)
  - Basic probability theory
  - Sums
  - (lots of) Algebra
- Algorithms
  - Basic data structures (lists, queues, trees, graphs)
  - Basic algorithmic techniques (loops, conditions, recursion)
- Any gaps? Start filling immediately!
  – See textbook appendix A, B, C
  – Read Chapters 1,2,3 and 4 of text book (Quiz 1 on Jan. 16)
  – Additional handouts will be posted if necessary

Refreshing your Math

- Textbook appendices
- “Mathematics for Computer Science” course on MIT Open Courseware
  - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-spring-2010/
- “Mathematics for Computer Science” notes by MIT
  - Posted on Moodle (https://source.masdar.ac.ae)

Programming

- This is a “theory” course
- Programming can help your understanding
- There will be some programming in the homeworks
- All programs are to be written in Python

One-off Python overview session:

- **Time:** Tuesday, 14 Jan, 11:00-12:00 am
- **Place:** IT Lab, Level 3
- **Demonstrator:** Mukesh Jha

Why Algorithms?

- Single most important course in a CS degree
- Essential for interviews with serious companies:
  - Google
  - Microsoft
  - Facebook
  - …

Let's Explore!

Analysis of algorithms

The theoretical study of computer-program performance and resource usage.

What's more important than performance?

- modularity
- correctness
- maintainability
- functionality
- robustness
- user-friendliness
- programmer time
- simplicity
- extensibility
- reliability

Why study algorithms and performance?

- Algorithms help us to understand scalability.
- Performance often draws the line between what is feasible and what is impossible.
- Algorithmic mathematics provides a language for talking about program behavior.
- The lessons of program performance generalize to other computing resources.
- Speed is fun!

The problem of sorting

Input: sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of numbers.

Output: permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ Such that $a'_1 \leq a'_2 \leq \ldots \leq a'_n$.

Example:

Input: 8 2 4 9 3 6

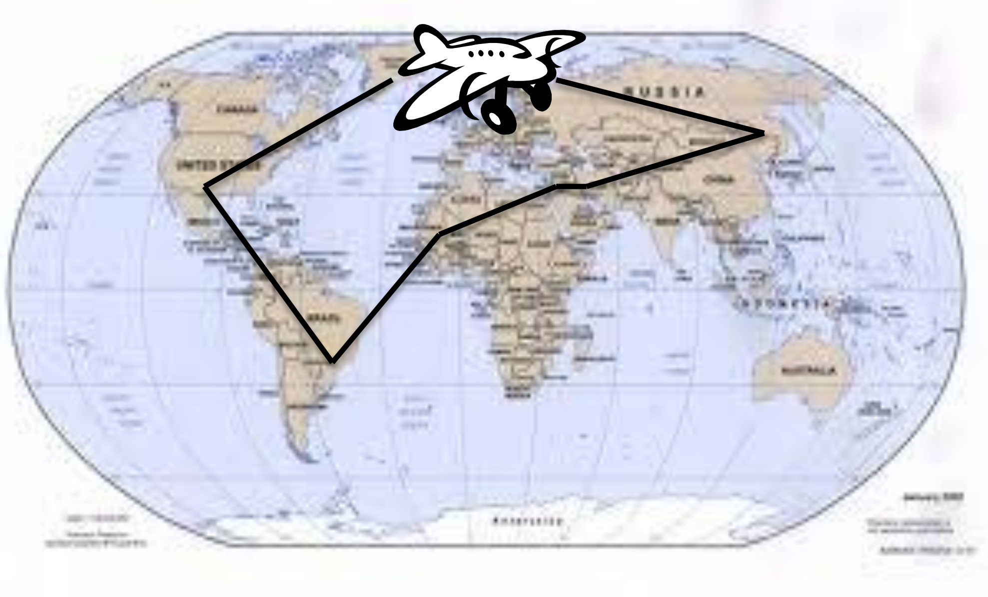
Output: 2 3 4 6 8 9

The TSP problem

(Travelling Sales Person problem)

Input: sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of cities with pairwise distances c_{ij}.

Output: permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ that minimizes the total distance for visiting all cities in that order and coming back home.



Simple exhaustive search

Try all permutations

Running time is proportional to $n! \sim n^n$

Suppose that n=20

Then on your laptop (say even with 200,000 MIPS), you need typically

$$\frac{20^{20}}{200{,}000 \times 10^6 \times 3600 \times 24 \times 365} \sim 16625063 \text{ years}$$

The TSP problem

(<u>T</u>ravelling <u>S</u>ales <u>P</u>erson problem)

Bad news: you will get $1000,000 if you get a "good" algorithm to solve the problem optimally.

Good news: you can approximate the optimal solution in reasonable time.

The problem of sorting

Input: sequence $\langle a_1, a_2, \ldots, a_n \rangle$ of numbers.

Output: permutation $\langle a'_1, a'_2, \ldots, a'_n \rangle$ Such that $a'_1 \leq a'_2 \leq \ldots \leq a'_n$.

Example:

Input: 8 2 4 9 3 6

Output: 2 3 4 6 8 9

Two algorithms

- *Insertion sort*
- *Merge sort*

- *Main question*: Which one is better?

Two algorithms

- *Insertion sort* O(n^2)
- *Merge sort* O($n \log n$)

- *Main question*: Which one is better?