ALGORITHMS

Lecturer (that's me): **Gabriel Istrate**

- My office: e-Austria Research Institute, 045B (next door)
- http: //gabrielistrate.weebly.com
- e-mail: gabriel.istrate@gmail.com
- Schedule:

  Lecture: every Thursday at 13:00   (room 045C)

  Seminar: Thursdays, after lecture.

  Lab: Marian Neagul,marian@info.uvt.ro

What is this course about ?

Have you wondered how :

- search engines (e.g. Google) quickly return reasonably relevant answers to your queries ?
- Spam filters detect unwanted messages ?
- Travel sites e.g. travelocity, kayak, skyscanner give you travel options, including multi-leg trips ?

What is behind these tools?

- NOT ONLY PROGRAMMING !
- Naïve approaches won't work as fast/well as you want them to.
- (YES) SOME MATH

What is this course about ?

- What is behind these tools?
  - Algorithms for searching, keywords matching, sorting, frequency computation, correlations identification etc.

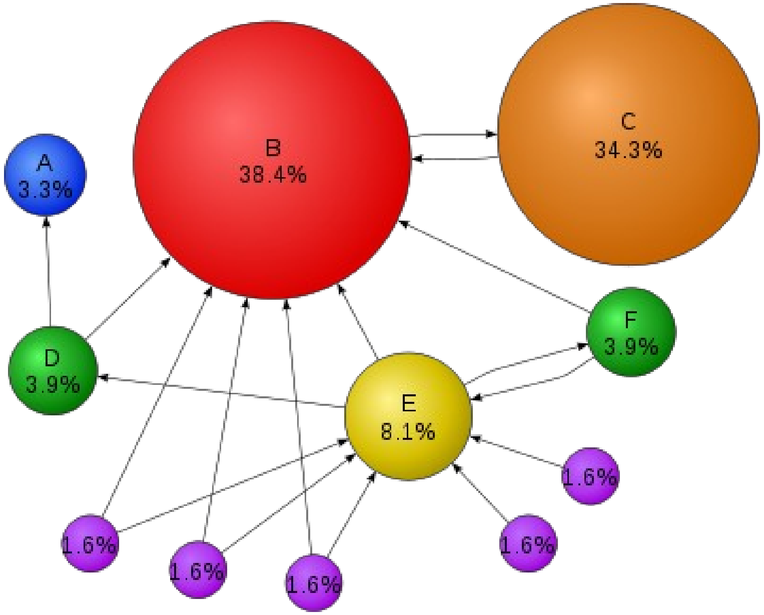
Examples: PageRank (Google), EdgeRank (Facebook), shortest path (Dijkstra, etc) - travel applications.

Analogy: cooking.

You all may have a basic idea on cooking. Being a “master chef” takes more know-how.

What is this course about ?

PageRank – algorithm used by the Google search engine to rank the web pages [Larry Page, 1997]



Basic idea of ranking:

$\text{rank}(P_0)=(1-d)+d*(\text{rank}(P_1)+\ldots\text{rank}(P_k))$

P_0 – current page

$P_1,\ldots, P_k$ – pages which contain links toward P_0

d in (0,1) – damping factor (models the influence of time)

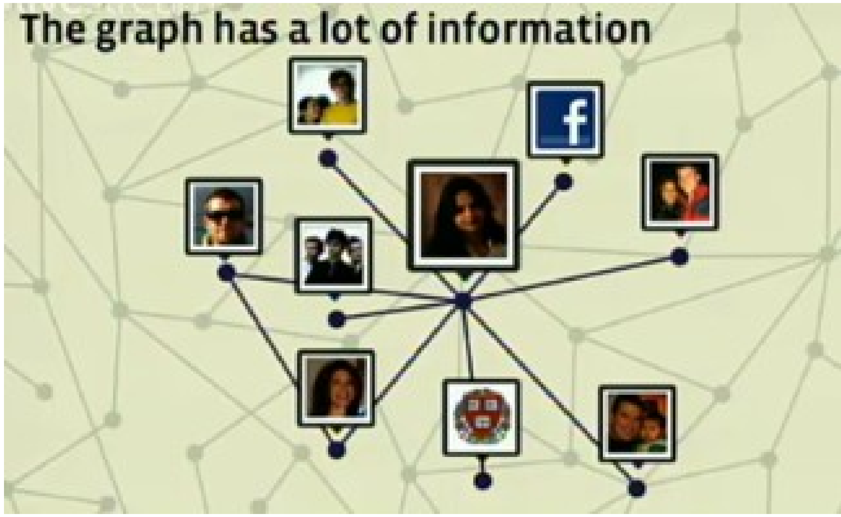
Web = graph

Ranking criteria = probabilistic scores

Rank computing = iterative algorithm or algebraic compution (solving as linear system)

What is this course about ?

EdgeRank – algorithm used in Facebook news feed (selection of news to be posted on the wall of a user)



http://techcrunch.com/2010/04/22/facebook-edgerank/

http://www.youtube.com/watch?v=kI4YIYInou0

Basic idea:

- The interaction between a user and a facebook "object" (e.g. info, comment etc) defines an edge
- Each edge: 3 "importance" factors: affinity (between the user and object creator), weight, age.
- More important edges are included in News with higher probability.

Algorithmics – more than a university course

Classical research area in Computer Science. Some modern directions :

- “Algorithmic game theory”
- “Algorithms related to social networks”
- “Online algorithms”
- “Streaming algorithms”
- “Randomized algorithms”

If you want to know more/do something beyond this course feel free to talk to me.

What is this course about ?

- This course is about:
  - designing and analyzing algorithms
  - abstract thinking and solving problems
- This course is NOT:
  - a programming course

    (however we will use **Python** (http://www.python.org)
  - a math course

    (however we need/use some basic math stuff: sets, function, relation; some combinatorics; some mathematical logic. Proof techniques e.g. mathematical induction or proof by contradiction)

Why such a course could be useful for you?

A computer scientist must be prepared for tasks like:

” … This is the problem. Solve it ...”

In such a situation it does not suffice to know
how to code a given algorithm

You must be able to find an adequate algorithm or even
develop a new one to solve the problem

Why such a course could be useful for you?

Technology changes. Some things are :

- Content: fundamental problems and solutions

- Method: Principles and techniques to solve the vast array of unfamiliar problems that arise in a rapidly changing field

Syllabus

Fourteen lectures on:

1. Introduction to algorithmic problem solving
2. Description of algorithms
3. Verification of algorithm correctness
4. Analysis of algorithm efficiency
5. Sorting and searching
6. Basic techniques in algorithm design:
   a) divide and conquer, decrease and conquer
   b) greedy
   c) dynamic programming
   d) backtracking, branch and bound

Course webpage

Web page: http://gabrielistrate.weebly.com/algorithms.html

lectures files,slides, exercises for seminar/lab

homework

pointers to other video courses on algorithms. Feel free to use alternate material !

If you find typos or other errors please let me know !

Course material

Based mainly on:

1. T.H. Cormen, C.E. Leiserson, R. Rivest, C. Stein - Introduction to algorithms, MIT Press, third edition 2009.

2. J. Kleinberg, E. Tardos – Algorithm Design, Addison Wesley, 2005

Various course slides/lecture notes due to

Daniela Zaharie (UVT)

Jeff Edmonds (York University, Canada),

David Luebke (Virginia University, USA),

Steven Rudich (Carnegie Mellon University, USA) ...

Rules & expectations

Homework should be returned by the due date; late homework does not count.

Course attendance: recommended, not enforced.

However seminars/labs are (by university policy) mandatory. More than 2 absences at seminar/lab lead to failing the course.

- Sharing ideas is fine, and encouraged. Sharing solutions solutions is not.
- Plagiarism of homework or written test is punishable by educational regulation and will be punished harshly.
- To my mind: university education not primarily about grading. But grades should reflect your effort
- "Good grades" will require significant amount of effort.

Rules & expectations

In return:

- approachable. Suggested call: “Gabi”.
- will try to respond to your questions. Convenient time for office hours ? Preferably not Monday/Friday, generally in the afternoon.
- will try my best to be fair/objective.
- I am teaching this course for the first time (was a TA for it in the States) Mistakes/typos may appear in slides. You're welcome to tell me about it (in public too)
- Welcome to interrupt me in class, provided you exercise judgement in using this (e.g. know required material when asking questions)
- Please let me know how I can make this more enjoyable/interactive.