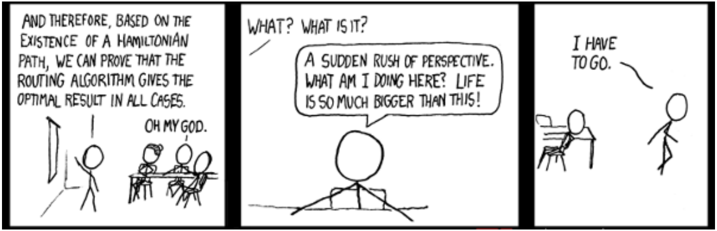
Algorithms (CC4010)

Pedro Ribeiro

DCC/FCUP

2018/2019


AND THEREFORE, BASED ON THE EXISTENCE OF A HAMILTONIAN PATH, WE CAN PROVE THAT THE ROUTING ALGORITHM GIVES THE OPTIMAL RESULT IN ALL CASES.
OH MY GOD.
WHAT? WHAT IS IT?
A SUDDEN RUSH OF PERSPECTIVE. WHAT AM I DOING HERE? LIFE IS SO MUCH BIGGER THAN THIS!
I HAVE TO GO.

Who am I?

Name: Pedro Ribeiro
Office 1.47 (FC6 - DCC building)
Personal Website: http://www.dcc.fc.up.pt/~pribeiro/

PhD in **Computer Science**

Main research interests:

- Algorithms and Data Structures, Complexity.
- Complex Network Analysis, Graph Mining, Data Mining.
- Parallel and Distributed Computing, Scheduling.
- Bioinformatics Applications; Biological Networks; Social Networks.

Other research interests:

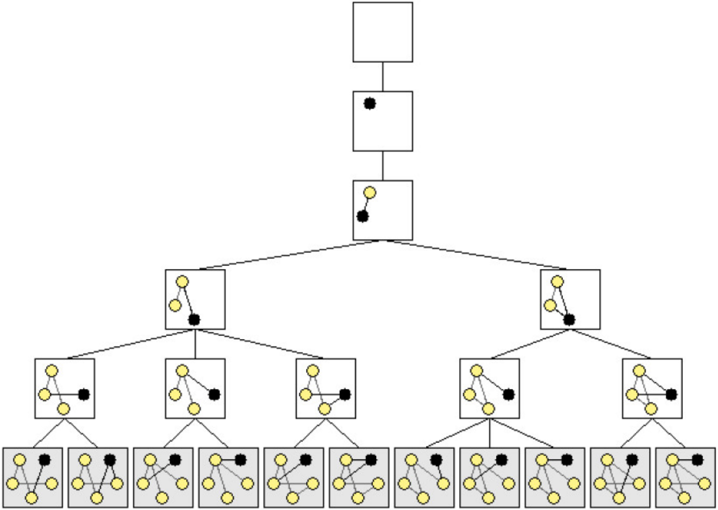
- Computer Science Education and Programming Contests.
- Artificial Intelligence, Agents, Machine Learning and Robotics.

PhD in Computer Science (2011)

Thesis: Efficient and Scalable Algorithms for Network Motifs Discovery

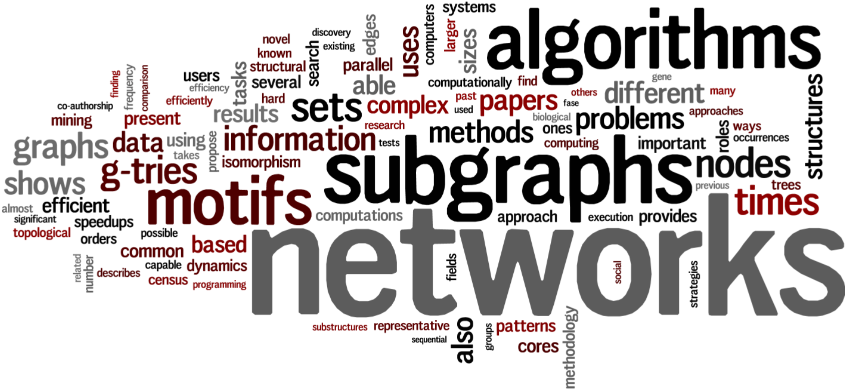
Efficient and Scalable Algorithms for Network Motifs Discovery
Pedro Ribeiro
Doctoral Programme in Computer Science
U.PORTO

PhD: G-Tries



Publications

A word cloud of my publications abstracts *(made in Dec 2015)*



http://www.dcc.fc.up.pt/~pribeiro/pubs_by_year.html

Competitive Programming

I'm involved on several types of **algorithmic programming contests**: *(organizing, creating and solving problems, training students, ...)*

- High-School Students
  - National and International Olympiads in Informatics
- University Students
  - ACM-ICPC national an international contests (e.g. MIUP, SWERC)

General Information

- **Site:** http://www.dcc.fc.up.pt/~pribeiro/aulas/alg1819/
- **Classes:**
  - **Tuesday:** 10:00-13:00, Room FC6 140 (S2)
- **Typical Class:**
  - First Part: theoretical exposition
  - Break: some time for resting, having a coffee...
  - Second Part: some more "practical" exercises
- **Class participation:** There is no minimum enforced attendance

Evaluation

- **(H) Written + Oral Homework:** 2 written assignments to be solved individually and submitted electronically + 2 assignments to be presented orally. H weight in your final grade: **30%**
- **(E) Final Exam**: 70% of your final grande
- **(T) Multiple smaller exams:** Two exams, **T1** and **T2** will be offered during the semester (one midterm and one near the end). You can use them to get exemption from the final exam.
  $T = 0.5 \times T1 + 0.5 \times T2$

1st Exam ("Normal") classification: $C = max(E, T) * 0.7 + H * 0.3 >= 9.5$
2nd Exam ("Recurso") classification: $C = E * 0.7 + H * 0.3 >= 9.5$

E, T1 and T2 should have a minimum grade of 8 out of 20.

Homework - Written Assignments

- Should be **submitted electronically by email**
- You do not need to deliver a typed version of you answers, but you should make sure they are **legible**. I recommend the usage of LaTeX .
- You may work "in group", but you should do your **individual writeup**. This means you can collaborate by talking about the problems, but you should not copy writeups. If you were in a group, please acknowledge that and list the other students with whom you talked about the problem.
- If you use any reference, webpage or solution from other classes, **you must cite it**.

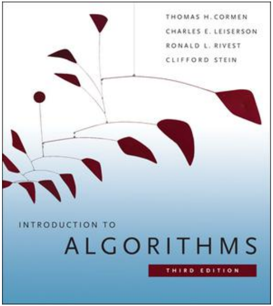
Homework - Oral Assignments

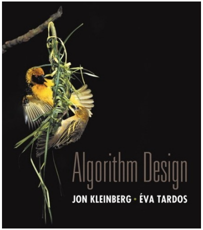
- **Groups of 3 students** and each assignment will have **3 problems**. You will work together to solve the problems (collaboration is required).
- You will **present your solutions to me**. Each group will have 30m time slots and at each presentation each member will spend 10m presenting one of the problems (I will decide whom presents what problem). While one student is presenting, the other may "intrude" a little bit.
- If you are really nervous about your presentation, you may hand in an extra written sketch of the solution for being taken in consideration
- **The three presentations will determine the score of the group**. I however reserve the right to give different grades to different members of the groups when I believe that is warranted.

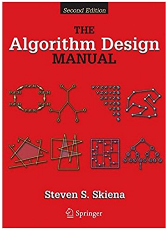
Learning outcomes this curricular unit

- Understanding the relationship between **algorithm design, correctness proof and complexity analysis**.
- Learning **algorithmic techniques** of general applicability.
- Knowing some **major algorithms** in a few common domains.
- **Practical experience** in applying generic algorithms to specific problems.

Recommended Books





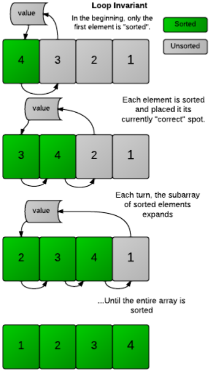


Competitive Programming 3
The New Lower Bound of Programming Contests
Steven Halim
Felix Halim
HANDBOOK FOR ACM ICPC AND IOI CONTESTANTS
2013

An overview of the program

Correctness and Loop Invariants:

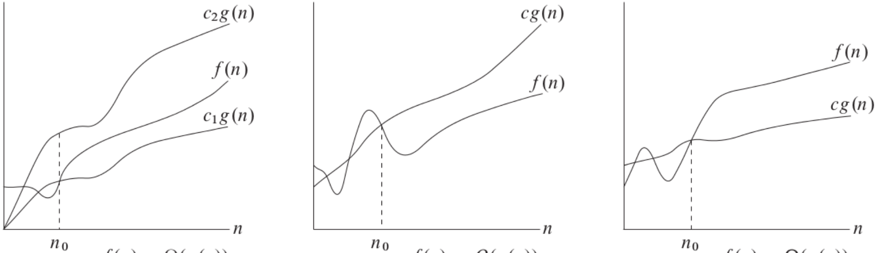
- The role of invariants
- Using invariants for proving and for designing new algorithms


Loop Invariant
In the beginning, only the first element is "sorted".
Sorted
Unsorted
value
4
3
2
1
Each element is sorted and placed it its currently "correct" spot.
value
3
4
2
1
Each turn, the subarray of sorted elements expands
value
2
3
4
1
...Until the entire array is sorted
1
2
3
4

An overview of the program

Asymptotic Analysis:

- Fundamentals
- Notation: O, Ω, Θ, o and ω
- Analyzing cycles and summations

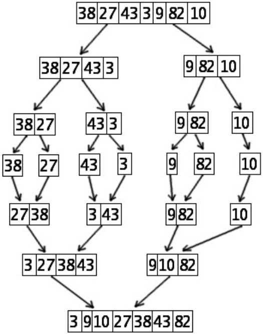


$f(n) = \Theta(g(n))$ $f(n) = O(g(n))$ $f(n) = \Omega(g(n))$

An overview of the program

Divide-and-Conquer Design Pattern

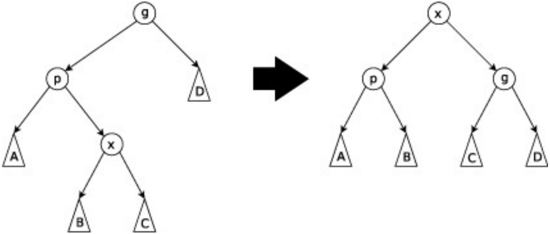
- Examples of usage
- Solving recurrences (unrolling, guess and proof, master formula))



An overview of the program

Amortized Analysis

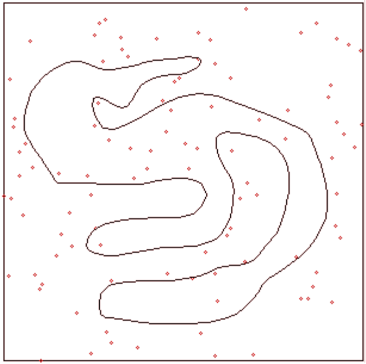
- Examples of usage
- Aggregation, accounting and potential methods



An overview of the program

Probabilistic Analysis

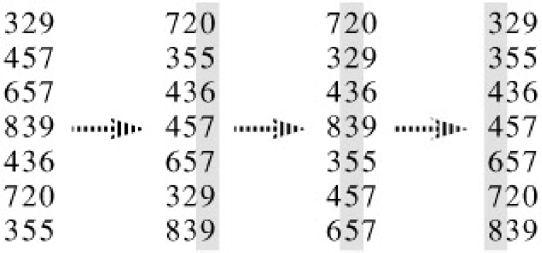
- Average-case analysis
- Randomized algorithms



An overview of the program

Revisiting sorting and selection

- Sorting/order concepts
- Bucket sort and radix sort
- Quickselect

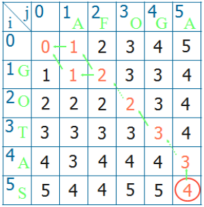


329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

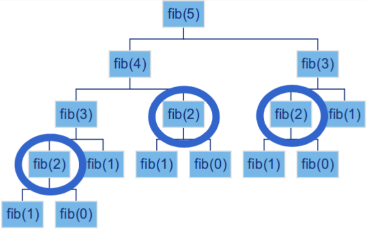
An overview of the program

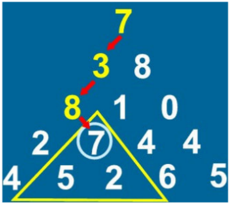
Dynamic Programming

- Optimal substructure and overlapping subproblems
- Top-down (memoization) and bottom-up strategies
- Some easy, medium and more advanced examples



i \ j	0	1 A	2 F	3 O	4 G	5 A
0	0	1	2	3	4	5
1 G	1	1	2	3	3	4
2 O	2	2	2	2	3	4
3 T	3	3	3	3	3	4
4 A	4	3	4	4	4	3
5 S	5	4	4	5	5	4

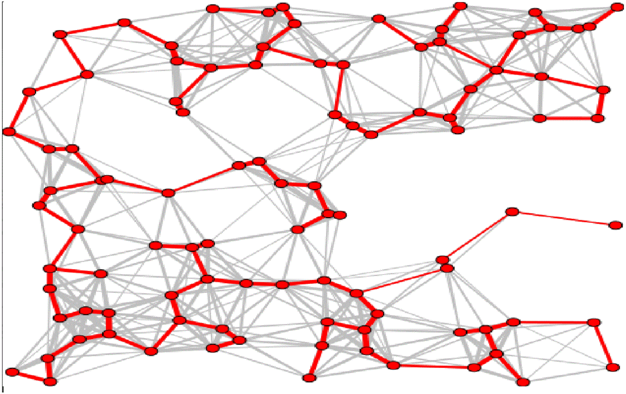




An overview of the program

Greedy Algorithms

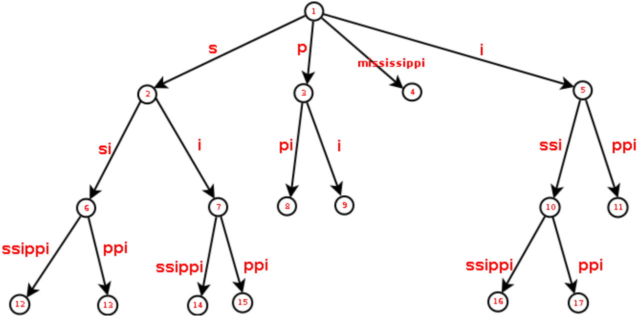
- Examples and some details about correctness
- Approximation algorithms



An overview of the program

String Algorithms

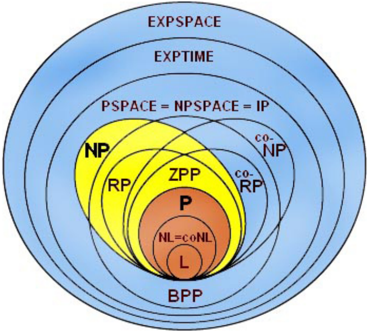
- String Matching (Knut-Morris-Pratt, Rabin-Karp, ...)
- Tries and Suffix Trees/Arrays



An overview of the program

Introduction to the theory of NP-completeness

- Classes of problems
- Reductions


EXPSPACE
EXPTIME
PSPACE = NPSPACE = IP
NP
co-NP
RP
ZPP
co-RP
P
NL=coNL
L
BPP