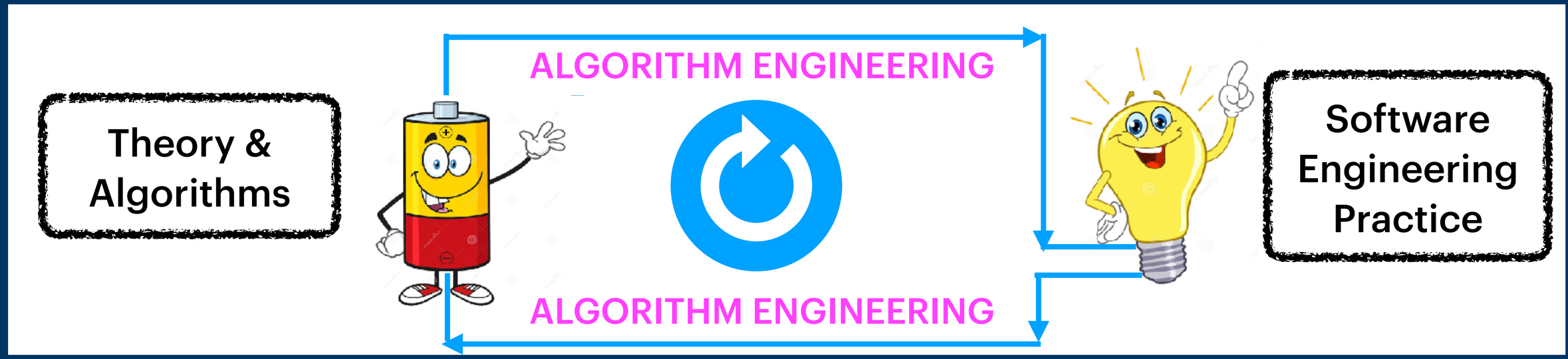


ALGORITHM ENGINEERING

Lecture 1:
Introduction to Experimental Algorithmics

M. Oğuzhan Külekci - kulekci@itu.edu.tr

What Would You Do When Assigned a Computational Task?

…particularly, a hard challenging one

Classical Approach
With Pen - Paper

1. Abstraction / Model of Computation
2. Algorithm/data structure design
3. Analysis of the solution

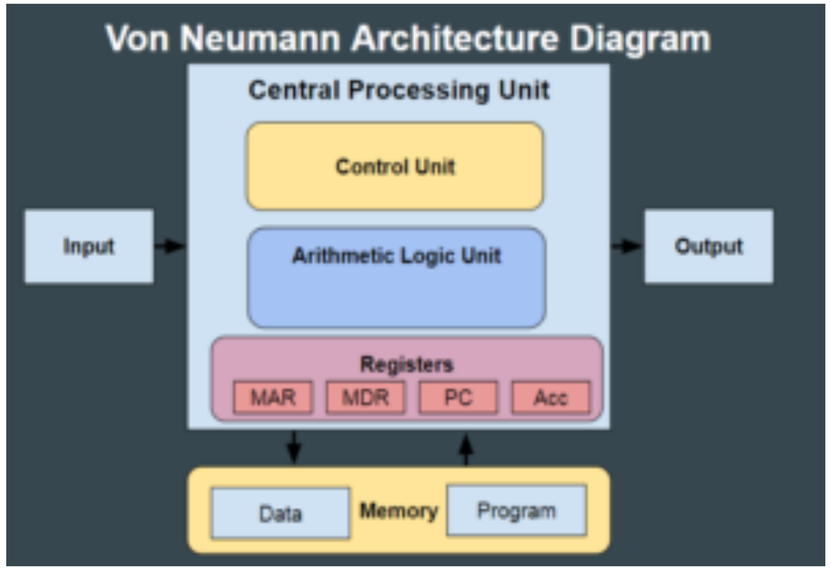
Experimental Approach
Real-world applications

4. Implemention
5. Experimentation

Experiments provide specify, but lacks generality !!!

Modeling

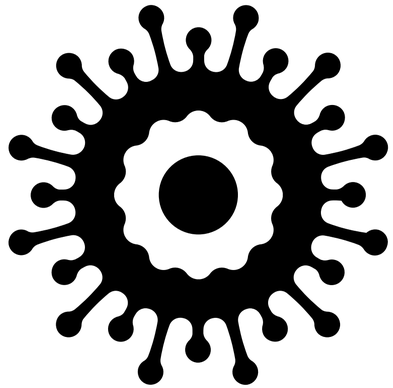
Modern Computing Architectures



Is it still realistic?

Consider the instruction parallelism, branch prediction, cache-effect, different memory architectures, connections, different distributed/parallel computing platforms, hardware properties, etc…

Real-world Problem Instances



Theoretical setting of the problems are too general with pessimistic worst-case assumptions.

Algorithm/Data Structure Design & Analysis

"Asymptotically better" not always implies "practically efficient"

- **Hidden constants**: Asymptotic notation obscures details
  - e.g., suffix trees occupy linear $O(n)$ space !
- **Elegance** : Complicated schemes make it difficult to implement
  - e.g., Path-ORAM versus Opt-ORAM
- **The effect of computation model** : Too general in classical approach
  - e.g., SIMD support for vectorization
- "**Efficient**" in theory means **poly-time**, which can be hard in practice even for small degrees
  - e.g., matrix multiplication $O(n^3)$

Implementation

GENERAL

Paradigms

Algorithm

Source Code

Object Code

Process

SPECIFIC

CORRECTNESS: Make sure the algorithm returns correct results,

- Analytical problems in the algorithm
- Numerical implementation faults

EFFICIENCY: Make it run fast with small resource usage.

For an efficient executable at the end, take care for

1. System structure: Interaction of sub-modules, sequential or parallel platform
2. Algorithm/data structure design : Devise/use most appropriate ones
3. Implementation and algorithm tuning : Tune the algorithmic details in implementation
4. Code tuning : Reduce instruction count or time per instruction
5. System software : Operating system, programming language, compiler
6. Platform and hardware : Better hardware/platform/library

Experimentation

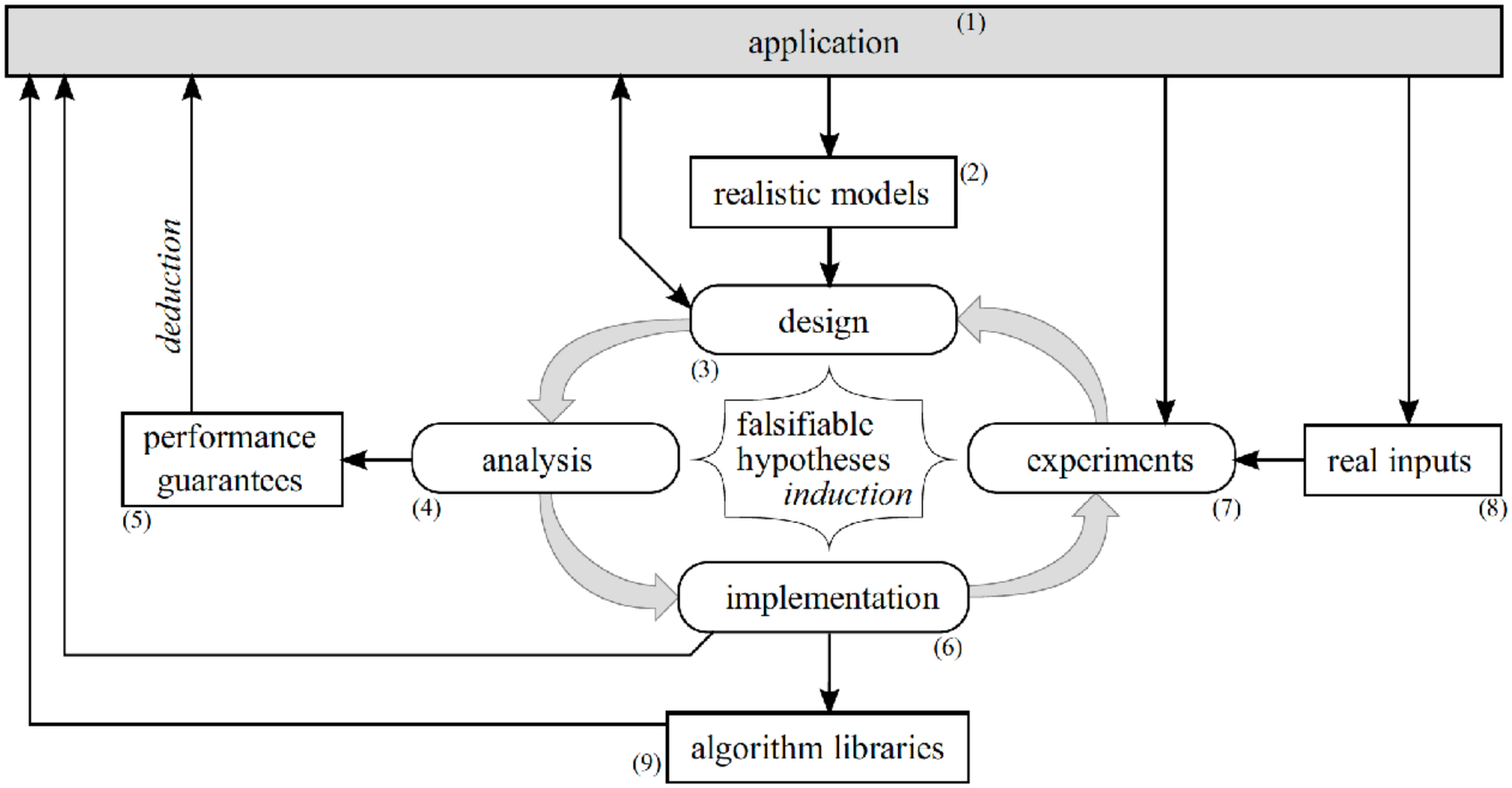
- **What do we want to observe, what is the aim/hypothesis ?**
- **What are the parameters to monitor ?**
- **How can we design the experiments for most clear answers?**

EFFICIENCY: Make it run fast with small resource

For an efficient executable at the end, take care for

1. System structure: Interaction of sub-modules, sequential or parallel platform
2. Algorithm/data structure design : Devise/use most appropriate ones
3. Implementation and algorithm tuning : Tune the algorithmic details in implementation
4. Code tuning : Reduce instruction count or time per instruction
5. System software : Operating system, programming language, compiler
6. Platform and hardware : Better hardware/platform

Algorithm Engineering Cycle



*Chimani, M., & Klein, K. (2010). Algorithm engineering: Concepts and practice. In *Experimental methods for the analysis of optimization algorithms* (pp. 131-158). Springer, Berlin, Heidelberg.

Algorithm Engineering

Purely empirical approach

Specific, but not general.

Purely theoretical approach

General, but no specificity.

Algorithm Engineering Approach

Systematic experimentation without omitting theory.
Theoretical thinking without neglecting real-world practice.

"Efforts must be made to ensure that promising algorithms discovered by the theory community are implemented, tested and refined to the point where they can be usefully applied in practice. [...] to increase the impact of theory on key application areas."
[Aho et al. (1997), Emerging Opportunities for Theoretical Computer Science]

"Algorithm Engineering is concerned with the design, theoretical and experimental analysis, engineering and tuning of algorithms, and is gaining increasing interest in the algorithmic community. This emerging discipline addresses issues of realistic algorithm performance by carefully combining traditional theoretical methods together with thorough experimental investigations."*(posted in DMANET on May 17, 2001 by Guiseppe Italiano, ALCOM-FT Summer School on Algorithm Engineering)*

"Algorithm Engineering refers to the process required to transform a pencil-and-paper algorithm into a robust, efficient, well tested, and easily usable implementation. Thus it encompasses a number of topics, from modeling cache behavior to the principles of good software engineering; its main focus, however, is experimentation." David Bader, Bernard Moret,Peter Sanders, SEA'2002

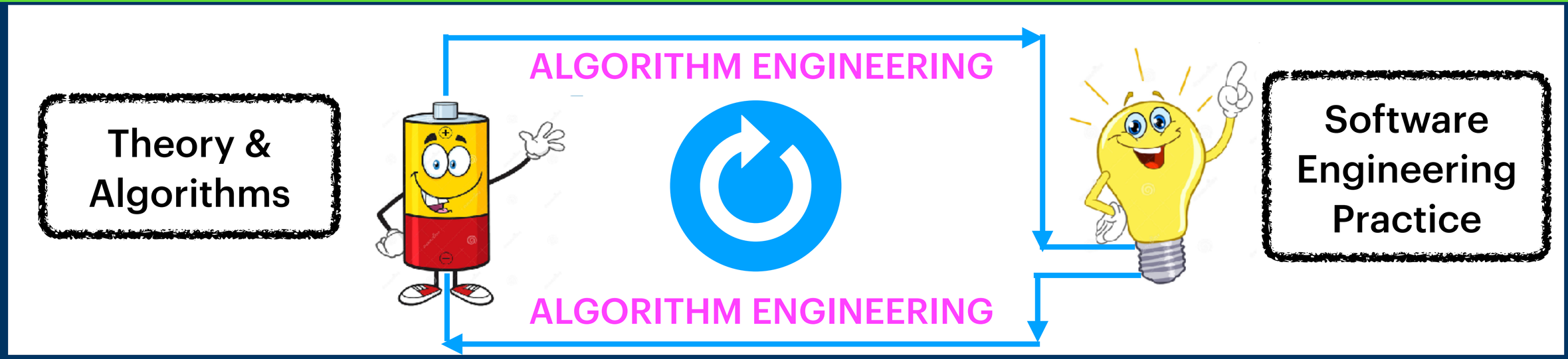
Algorithm Engineering

Reading Assignments

1. MCGEOCH, Chapter 1.
2. MS, Chapter 1.
3. BCPP, Chapter 6, Algorithm Engineering: Concepts and Practice, Markus Chimani and Karsten Klein

- Moret, B. M. (2002). Towards a discipline of experimental algorithmics. Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, 59,197-213. https://www.cs.unm.edu/~moret/dimacs_algorithmics.pdf
- McGeoch, Catherine C. "Experimental algorithmics." Communications of the ACM 50, no. 11 (2007): 27-31. https://cs.uwaterloo.ca/~brecht/courses/854-Experimental-Performance-Evaluation-2018/readings/experimental-algorithmics-cacm-2007.pdf
- Sanders, P. (2009). Algorithm engineering–an attempt at a definition. In Efficient algorithms (pp. 321-340). Springer, Berlin, Heidelberg. http://algo2.iti.kit.edu/documents/definition.pdf
- Snodgrass, R. T. (2011). On experimental algorithmics: an interview with Catherine McGeoch and Bernard Moret. Ubiquity, 2011(August), 1-14. https://ubiquity.acm.org/article.cfm?id=2015997

NEXT LECTURE ...



ALGORITHM ENGINEERING

Lecture 2:
Modeling from Algorithm Engineering Perspective

M. Oğuzhan Külekci - kulekci@itu.edu.tr

Course Outline

PREREQUIST

I WILL ASSUME

1. EVERYONE IN THIS CLASS HAS THE KNOWLEDGE OF BASIC **“DATA STRUCTURES AND ALGORITHMS“**
2. EVERYONE IS FLUENT ON A **PROGRAMMING LANGUAGE**, PREFERABLY C/C++, BUT JAVA, PYTHON ARE ALSO FINE

IF YOU FAIL IN THE ABOVE PREREQUISTS, THEN IT WILL BE EXTREMELY DIFFICULT TO FOLLOW !

Course Outline

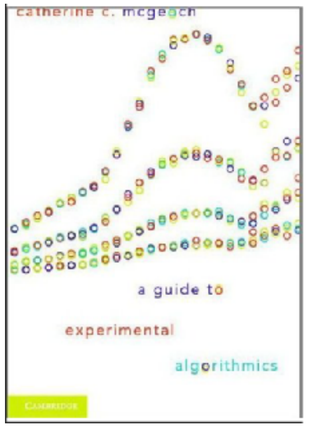
Weekly Plan

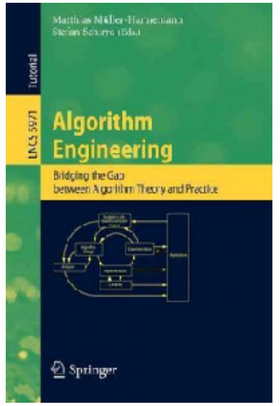
COURSE PLAN

Weeks	Topics
1	The concept of algorithm engineering
2	Modeling the problem and realistic computing models
3	Case studies with student presentations on modeling and computing models
4	Design phase: Design of experiments
5	Design phase: Measuring the performance
6	Algorithm analysis phase
7	Case studies on design and analysis phase with student presentations
8	Implementation phase - Techniques to reduce the instruction count
9	Implementation phase - Memory and I/O optimization to reduce instruction time
10	Implementation phase - Concurrency and parallelism to reduce instruction time
11	Implementation phase - Software optimization and libraries in algorithm engineering
12	Testing phase – Issues in correct performance monitoring on correctly generated test data
13	Case studies with student presentations on design and analysis phase
14	Case studies with student presentations on reducing instruction time

Course Outline

Text Books





MAIN TEXT BOOK :

1) (MCGEOCH) McGeoch, C.C., A guide to experimental algorithmics, Cambridge University Press, 2012

WE WILL BE FREQUENTLY READING FROM:

2) (MS) Müller-Hannemann, M., Schirra, S., Algorithm Engineering, Springer, 2010

ADDITIONAL BOOKS:

3) Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M., Experimental methods for the analysis of optimization algorithms, Springer, 2010
4) Kliemann, L., Sanders, P., Algorithm Engineering: Selected Results and Surveys. Springer, 2016

FURTHER DOCUMENTS AND PAPERS MAY BE ASSIGNED FOR READING DURING THE LECTURES

Course Outline

ASSESSMENT

Başarı Değerlendirme Sistemi (Assessment Criteria)	Faaliyetler (Activities)	Adedi* (Quantity)	Değerlendirmedeki Katkısı, % (Effects on Grading, %)
	Yıl İçi Sınavları (Midterm Exams)		
	Kısa Sınavlar (Quizzes)	4	%20
	Ödevler (Homework)	4	%20
	Projeler (Projects)		
	Dönem Ödevi/Projesi (Term Paper/Project)	1	%20
	Laboratuvar Uygulaması (Laboratory Work)		
	Diğer Uygulamalar (Other Activities)		
	Final Sınavı (Final Exam)	1	%40