Mathematics for Computer Science: Math applications in design and analysis of algorithms

January - March 2020

Organizer:	ACM UTEC	**Time:**	XX:00 – XX:00
Email:	acm@utec.edu.pe	**Place:**	Unconfirmed @UTEC

1 Course Pages

All resources used through the camp can be found in the following Github repository:

- https://github.com/acm-utec/CS2102-training-camp

2 Schedule

Three to four videos per week accompanied with two three-hour reunions per week throughout the 2020-0 academic period. Specific tasks per class are found in Appendix N1. The desirable distribution of the 3 hours is one hour to discuss and talk about the information retrieved from the videos and two hours to try and solve problems.

3 General objectives

- Apply knowledge acquired from previous Mathematics for Computer Science and Data Structures courses to improve students' abilities in Design and Analysis of Algorithms, being able to analyze the performance of algorithms, write coherent proofs for algorithms, demonstrate familiarity with various algorithms and data structures, and apply algorithmic designs and methods of analysis. This would give the student the opportunity to have better performance in class or/and keep improving by himself with practice.
- Fill the knowledge gaps in students that have already taken the course.
- Raise awareness of the importance of some courses in Computer Science (CS1D01, CS1D02, CS2101, CS2102).
- Emphasize the significance of hard work, curiosity and intellectual vitality.
- Promote the elaboration and organization of study groups among peers.

4 Prerequisites

No specific requirements are necessary, just the desire to learn. However, a good understanding of CS1D01, CS1D02, and CS2100 are highly recommended, as many of the study content will assume complete knowledge from these courses, and will not go over those topics in detail. Bear in mind, however, that Mathematics are key in the entirety of Computer Science so strengthening these skills will be of great use in the course of the following years.

5 Methodology

We **will assume previous knowledge** on the courses CS1D01, CS1D02, and CS2100. Nevertheless, we encourage anyone with the preparation and desire to learn from these topics to participate, as learning is always good. Students will need to *learn to learn* in new ways, as with **no one tutoring them**, but rather guiding themselves and other assistants through the development of the topics. This will have a fair level of difficulty, but what is important is to never give up in the face of adversities. You will always have resources, friends, and peers around to help and make mistakes along with you, but trying is crucial to build ourselves as good professionals.

We understand that 8 weeks is a very short time to cover so many complex topics with the aforementioned assumptions and the level of comprehension desired, so we're tackling this problem by taking advantage of two things: 1) the *unique experience* of individuals with different learning strategies and resources, and 2) the *exchange of ideas* between those unique individuals. In order to maximize these effects, the class will also be split in groups. The reason for this decision is that sometimes the amount of people that participate in exchange of ideas in traditional classes is very small. Those people tend to get a fairly good understanding by actively learning, but the remaining people don't. Dividing a class in small groups can lessen the pressure of asking questions or expressing one's own view while also simplifying the group dynamic, since helping a small group is much more manageable than helping a big one.

We will use varied means to learn, as using books, papers, and practice material, but our principal source will be MIT OpenCourseWare, from where we will review the Course Features (Resource A) provided for the 6.046J course, Design and Analysis of Algorithms.

These videos will be for the student to watch from home, alone or with friends, so that they can go on, pause, repeat, or inquire further on any topics they find interesting or hard to understand (We strongly encourage you to take notes of each of these things, so that you can share it with your peers). You will be expected to have seen between three or four videos weekly. On top of that, we will have two weekly meetings 3 hours long, where students will use the first hour to discuss and share ideas from what they learned from watching the videos; and the two final hours to gather and solve exercises in groups, showing in practice what they have learned. It is highly recommended that students spend at least 30-45 minutes trying to solve problems proposed before the meeting.

To make the most out of this study group, students must share their doubts, questions, and difficulties just as they must share their knowledge, understandings, and tips. Don't be shy, as progress is built by union. **Your perspective is important**.

We will appeal to videos and books to make sure we have multiple ways to learn. The videos used will be retrieved from "MIT 6.046J Design and Analysis of Algorithms, Spring 2015" found in Resources. From this point, we are going to refer to this set of videos as "MIT lessons", which playlist is Resource B. When referring to videos, we do not include Recitations, though it is suggested for you to check them. Remember, this is the minimum we expect students to work with, but you could use other resources and explore in other bibliography to inquire deeper on certain topics you desire to understand better or have trouble understanding.

Regarding the books, any student can use the books he finds useful, but a group of suggested groups are included in Resource D, and are referred to in each week's sources.

Videos listed must be watched and analyzed before each session. In the following week-by-week description, we specify which videos should be watched per week, while you will find which videos are expected that you watch before each class in the appendix Number 1.

6 Course Outline

Week I	
• Introduction to DAA. Brief recall of P, NP, and NP-Complete complexities, greedy algorithms and intractability, as well as polynomial operations and their representations. Application of Divide and Conquer.	• Must have watched videos 1 to 3 from MIT lessons. • To start, we will account the assistants, get familiarized with the workflow, and discuss the first couple of videos. Also, we will work in Problem Set #1 & #2.
Readings: [RS09] (Ch. 2, 3, 4, 30), [GT09] (Ch. 1, 5, 12), [TAR83] (Ch. 1)	

Week II	
• Keep working with Divide and Conquer Learn and analyze the van Emde Boas Tree. Introduction to Amortization and Randomized Algorithms.	• Must have watched Videos 4 to 6 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #3 & #4
Readings: [RS09] (Ch. 5, 17, 20), [GT09] (Ch. 1)	

Week III	
• Keep working with Randomization. Design and Analysis of Skip Lists, concepts of Hashing, and Augmentation of Data Structures	• Must have watched Videos 7 to 9 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #4 & #5
Readings: [RS09] (Ch. 5, 11, 14), [GT09] (Ch. 2, 3)	

Week IV	
• Start getting a deeper notion on Dynamic Programming, and using multiple algorithms to solve the same or similar problems. Reviewing Greedy Algorithms.	• Must have watched Videos 10 to 12 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #5 & #6
Readings: [RS09] (Ch. 15, 16), [GT09] (Ch. 5, 7)	

Week V	
• Learn about Network Flow, it's applications, and have an insight into Linear Programming.	• Must have watched Videos 13 to 15 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #5 & #6
Readings: [TAR83] (Ch. 9), [RS09] (Ch. 29), [GT09] (Ch. 8)	

Week VI	
• Review analysis of complexity of algorithms (P, NP, NP-Completeness and reductions). Learn about the use of Approximation Algorithms and Fixed-Parameters Algorithms.	• Must have watched Videos 16 to 18 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #7 & #8
Readings: [RS09] (Ch. 34, 35), [GT09] (Ch. 13)	

Week VII	
• Study of Distributed Algorithms, Synchronous and Asynchronous distributed algorithms, along with an introduction to Cryptography through Hash Functions.	• Must have watched Videos 19 to 21 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #8 & #9
Readings: [TAR83] (Ch. 6, 7) , [RS09] (Ch. 23, 22), [GT09] (Ch. 10)	

Week VIII	
• Conclude Encryption analyzing Cryptography, along with a deep analysis of Cache-oblivious Algorithms.	• Must have watched Videos 22 to 24 from MIT lessons. • We will discuss the next couple of videos. Also, we will work in problem Set #9 & #10
Readings: [GT09] (Ch.10), [COA12], [COB]	

Week IX	
• Finalize the training camp by solving tests and exams published in the OpenCourseWare.	• Must have finished all videos and practiced the exams beforehand. • In groups, we will all settle to solve the exams entirely.
Readings: [RS09], [GT09], [SF13], [TAR83], [COA12], [COB]	

7 Class Policy

- Full attendance is not expected, but is desirable. This is a fast-paced study group and although we care about the full understanding of each and every member, we won't hold back for those that don't come.

8 Main References

This is only a limited list of various interesting and useful books and online resources that will be touched upon during the course. You should consult them periodically. They are in order of relevance in the course syllabus structure, but each and every one of them provides rich insight into the topics.

8.1 OpenCourseWare Program

- Erik Demaine, Srini Devadas, and Nancy Lynch. 6.046J Design and Analysis of Algorithms. Spring 2015. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA. Recovered at: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/#

8.2 Books

`[NOTE:all books can be found in the Github repository!]`

[RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. Introduction to Algorithms, Third Edition. 3rd. The MIT Press, 2009

[GT09] Michael T. Goodrich and Roberto Tamassia. Algorithm Design: Foundations, Analysis and Internet Examples. 2nd. John Wiley & Sons, Inc., 2009

[SF13] R. Sedgewick and K. Wayne. Algorithms. Pearson Education, 2011.

[TAR83] Robert Endre Tarjan. Data Structures and Network Algorithms. Society for Industrial and Applied Mathematics, 1983

8.3 Camp material (Problem sets and exams per week):

- https://github.com/acm-utec/CS2102-training-camp/tree/master/resources/Camp%20Material

8.4 Papers

COA12 Frigo, M., Leiserson, C. E., Prokop, H., and Ramachandran, S. ACM Trans. Algor. 8, 1, Article 4 (January 2012), 21 pages. Retrieved from: http://supertech.csail.mit.edu/papers/FrigoLePr12.pdf

COB Michael A. Bender, Erik D. Demaine, Martin Farach-Colton. Cache Oblivious B-Trees, 18 pages. Retrieved from: https://erikdemaine.org/papers/CacheObliviousBTrees_SICOMP/paper.pdf

8.5 UTEC syllabi for every course

- Ernesto Cuadros. Compendio de Sílabos 2017, Escuela Profesional de Ciencia de la Computación. Recovered at: https://clei2004.spc.org.pe/Peru/CS-UTEC/Plan%202017/BookOfSyllabi-EN.pdf

8.6 Extra resources

- HackerEarth. Introduction to Dynamic Programming 1. Retrieved from: https://www.hackerearth.com/practice/algorithms/dynamic-programming/introduction-to-dynamic-programming-1/tutorial/
- Noteworthy - The Journal Blog. Top 50 Dynamic Programming Practice Problems. Retrieved from : https://blog.usejournal.com/top-50-dynamic-programming-practice-problems-4208fed71aa3

9 Collaborators

- Stephano Württele, `swurtteleigari@acm.org`
- Diego Linares, `dlinares@acm.org`
- Giordano Alvitez, `giordano.alvitez@utec.edu.pe`

Special thanks to PhD. Nancy Camarena Espinoza for all the feedback that went into the creation of this work.

10 Appendix

Daily Schedule			
Class	Date	Videos for the class	Problem Set to work
1	XX/1/20	Lessons 1 and 2	Problem Set #1
2	XX/1/20	Lesson 3	Problem Set #1 and #2
3	XX/1/20	Lessons 4 and 5	Problem Set #3
4	XX/1/20	Lesson 6	Problem Set #3 and #4
5	XX/1/20	Lessons 7 and 8	Problem Set #4
6	XX/1/20	Lesson 9	Problem Set #4 and #5
7	XX/1/20	Lessons 10 and 11	Problem Set #5
8	XX/1/20	Lesson 12	Problem Set #5 and #6
9	XX/2/20	Lessons 13 and 14	Problem Set #6
10	XX/2/20	Lesson 15	Problem Set #6 and #7
11	XX/2/20	Lessons 16 and 17	Problem Set #7
12	XX/2/20	Lesson 18	Problem Set #7 and #8
13	XX/2/20	Lessons 19 and 20	Problem Set #8
14	XX/2/20	Lesson 21	Problem Set #8 and #9
15	XX/2/20	Lessons 22 and 23	Problem Set #9
16	XX/2/20	Lesson 24	Problem Set #9 and #10
17	XX/3/20	All, should practice a lot before class	Tests and Exams
18	XX/3/20	All, should practice a lot before class	Tests and Exams