Information for Potential Students

August 10, 2024

Duc A. Hoang

Department of Informatics
Faculty of Mathematics, Mechanics, and Informatics
VNU University of Science, Hanoi, Vietnam
hoanganhduc@hus.edu.vn

A Brief Introduction
A Short CV

- **Name (in Vietnamese):** Hoàng Anh Đức
- **Name (in publications):** Duc A. Hoang
- **Current Position:** Lecturer at VNU-HUS, Hanoi, Vietnam (Feb. 2023 - present)
- **Research Interests:** Graph Algorithms, Combinatorial Reconfiguration
- **Education:**
  - **B.Math** degree from VNU-HUS, Hanoi, Vietnam (2008–2013)
  - **M.S.** and **Ph.D.** degrees (Information Science) from JAIST, Ishikawa, Japan (2013–2015 and 2015–2018 respectively)
- **Homepage:** `https://hoanganhduc.github.io/` (contains everything about my research and teaching)

Note

This document is intended for those considering working with me. Some information here may be useful for students in general

A Brief Introduction
Contact Me

If you want to work with me on some research problems:

- Please *skim through my recent publications and my list of participated events* to have some ideas of what I have been working on recently
  - I am happy to work on problems related to graph theory. *If you have some graph problems in mind that you are interested in and want to solve, I am happy to discuss with you to see if we can further collaborate*
- Please *read this document* to have some ideas (of what I have learned and collected so far) about doing research in TCS
- Please contact me by *sending an email* to my current email address
  - If we have not yet known each other, please *attach to your email a copy of your academic transcript* (an unofficial copy is fine) and *your CV* (both in **PDF format**, either in **English or Vietnamese**)
- I expect you to *at least* have *some basic knowledge on discrete mathematics and graph theory* (which can be obtained by taking an undergraduate-level course related to these subjects)
  - Please take a look at *some recommended materials in this document*. Vietnamese students can also look at some of my teaching materials at `https://hoanganhduc.github.io/teaching/`
  - Some other resources I collected are available at `https://hoanganhduc.github.io/misc/`

Publishing in TCS

- **(Most Important Point)** (Collaborate,) work hard, and achieve your results
- **(Optional)** Maintain an e-print version on arXiv (`https://arxiv.org/`)
- **(Optional)** Announce your results in some seminars, workshops, or non-refereed conferences
- **(Recommended)** Submit your results to a refereed conference
  - Some conferences are highly selective, for example, STOC, FOCS, SODA, and so on
  - Quick notification (accept/reject) within around three to six months
  - Usually having just one round of review. Reviewers have no idea whether their comments have been addressed by the authors. (Some conferences have "rebuttal phase")
  - Reviewers are primarily focused on whether the work is important and superficially appears correct
- **(Recommended)** Submit your results to a refereed journal
  - It is common to expand your conference's paper to a journal version. (Yes! You can publish both conference and journal versions of the same results)
  - Slow notification (accept/reject) within six months to a year
  - Your results will usually be reviewed thoroughly by 2-3 reviewers. Usually having more than one round of review
  - The journal version of your paper is the final version and can be trusted. (In several cases, flaws exist even in the journal version)

Things You May (Not) Already Know

- Traditionally, in most areas of mathematics and theoretical computer science, authors are listed in alphabetical order by last name
- Academic Profiles: Google Scholar, ORCID
- Mailing Lists: DMANET, THEORYNT
- Journal Ranking:
  - Scimago Journal & Country Rank
  - Scopus Indexed Journals
  - WoS (Web of Science) Indexed Journals
- Conference Ranking:
  - CORE Rankings Portal
  - Conference Ranks
- List of TCS conferences and workshops @ StackExchange
- Links to Combinatorial Conferences (maintained by Douglas B. West)

Things You May (Not) Already Know (cont.)

- Conferences in Theoretical Computer Science, maintained by Miki (Nicolas) Hermann
- Some conferences:
  - **(Top-tier)** STOC, FOCS, SODA
  - STACS, SoCG, ICALP, WG, ISAAC, ESA, MFCS, COCOON, FSTTCS, FCT, GD, CanaDAM
  - SWAT, WADS, IWOCA, IPEC, EUROCOMB, FUN, CCCG, EuroCG, TAMC, SOFSEM, WAOA, COCOA, LATIN, LAGOS, ITCS
  - CIAC, WALCOM, CALDAM
  - JCDCG^3, SEICCGTC, SOSA, HALG, BCC, MCCCC, ACC, AAAC, WAAC, DMD
- The Elsevier boyscott
- Uploading a paper to arXiv.org
- Online Collaboration
  - Online LaTeX editor: Overleaf

Things You May (Not) Already Know (cont.)

- Workspace: Slack, Zulip (support LaTeX), Discord, Google, Zalo (Vietnamese)
- Video Conference: Google Meet, Zoom, Jitsi Meet

An Inspiring Quote from Feynman

"You ask me if an ordinary person—by studying hard—would get to be able to imagine these things like I imagine. Of course. I was an ordinary person who studied hard. There's no miracle people. It just happens they got interested in this thing, and they learned all this stuff. They're just people. There's no talent or special miracle ability to understand quantum mechanics or a miracle ability to imagine electromagnetic fields that comes without practice and reading and learning and study. So if you take an ordinary person who's willing to devote a great deal of time and study and work and thinking and mathematics, then he's become a scientist."

— Richard P. Feynman

Richard Feynman: Fun to Imagine (BBC Series, July 1983)

https://youtu.be/nYg6jzotiAc&t=3301

David Eppstein's Two Models of Algorithms Research

Two Models of Algorithms Research

I. Read lots of theory papers

II. Choose a problem with lots of previous work (evidence it's interesting)

III. (optional) Add extra complications to the problem so you can convince people your results are more difficult than previous work

III. Find an algorithm that's better than all the previous results

IV. Write it up and publish it in theory conferences and journals

I. Learn about areas outside of theoretical CS

II. Choose a problem in one of those application areas where faster or more accurate solutions can make a practical difference

III. Abstract essential features to get new clean theoretical problem

IV. Find an algorithm that's better than all the previous results

V. Write it up and publish it in theory conferences and journals

VI. Implement and communicate your results with the community your problem came from, discover related problems, repeat

Open problems in graph theory and geometry — D. Eppstein, ICS 269, 01/25/02

Some Recommended Materials

Discrete Math, Graph Theory

- Discrete Mathematics and Its Applications, 8th edition, by Kenneth H. Rosen
- Building Blocks for Theoretical Computer Science, by Margaret M. Fleck
- Lectures on Discrete Mathematics given by Shai Simonson at ArsDigita University in 2000
- Connecting Discrete Mathematics and Computer Science, by David Liben-Nowell. A preprint version of the book is available
- Graph Theory, by Reinhard Diestel (GTM 173, 5th edition, Springer, 2016). The main text of the book can be freely viewed online
- Introduction to Graph Theory, by Douglas B. West (2nd edition, Prentice Hall, 2001)

Some Recommended Materials

Algorithms, Computational Complexity

- Computers and Intractability: A Guide to the Theory of NP-Completeness, by Michael Garey and David S. Johnson – One of the most influential books on the NP-complete theory, which is usually known as “the Garey&Johnson book”
- MIT 18.404J, Fall 2020, Theory of Computation, by Michael Sipser. (See this page for the PowerPoint slides.)
- Algorithms, by Jeff Erickson
- Computational Complexity: A Modern Approach, by Sanjeev Arora and Boaz Barak. A draft of the book is available
- Parameterized Algorithms by Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabhs

Some Recommended Materials
Writing

- Paul R. Halmos's advice on How to write Mathematics. Download a PDF copy here
- Scientific Paper Writing: A Survival Guide, by Bodil Holst, illustrated by Jorge Cham of PhD Comics

Some Recommended Materials

Advice/Inspiration

- Advice for solving graph theory problems by Matt DeVos
- Douglas B. West's Advice for students in Math 412
- List of proof techniques you should **not** use, by Dana Angluin (page 16)
- Gary MacGillivray's Math Study Tips
- Fan Chung Graham's A few words on research for graduate students
- Terrence Tao's career advice
- Mihir Bellare's The Ph.D Experience
- Ravi Vakil's advice For potential Ph.D. students
- Adrian Bondy's Beautiful conjectures in graph theory
- A Student's Guide to the Study, Practice, and Tools of Modern Mathematics, by Donald Bindner and Martin Erickson

Getting Scientific Papers/Books/etc.

- **(For Vietnamese)** The Online Portal maintained by Vietnamese National Agency for Science and Technology Information
- If you cannot get a paper, **get in touch with the (corresponding) author(s) to ask for a copy before trying the following resources**
  - Mutual Aid-Science Community – you may get published papers by asking people in the community for help
  - Sci-Hub
  - Z-Library
    - Desktop App: Windows, MacOS, Linux (DEB)
  - Library Genesis: here or here
  - Anna's Archive
  - PDFDrive