>>> Upgrading your memor[y|ies]

>>> Some guidance for UoA undergraduates considering postgraduate Computer Science

Name: James Cooper (School of Computer Science, University of Auckland)
Date: 20 August 2019

>>> Outline

1. Finding A Supervisor

2. 700 Prerequisites

3. Some Myths About Postgrad CS

4. Parting Words

Section 1

Finding A Supervisor

>>> Two approaches

You can either:

- Find a topic, and then a supervisor; or,
- Find a supervisor, and then a topic

All academic staff are expected to supervise some students, so if you approach them, they will listen. Just remember always to be respectful of them and their time, and don’t muck them about.

>>> First step

1. Figure out what fields of CS interest you most
2. Focus on one to three of them
3. Now either
   * Approach a potential supervisor who works in that field; or,
   * Continue narrowing your chosen field(s) to (a) specific topic(s)

>>> Topic before supervisor

If you want to find a topic, and then a supervisor to suit it.

There are (at least) three approaches:

- Think of a question. Keep reading until you find the answer. Repeat until you find a question that nobody knows the answer to.

You have found the beginning of your topic.

>>> Topic before supervisor

If you want to find a topic, and then a supervisor to suit it.

There are (at least) three approaches:

* Think of a question. Keep reading until you find the answer. Repeat until you find a question that nobody knows the answer to.

* Read in your chosen field until you come across a recent paper where you think ‘‘I could do that better.’’

You have found the beginning of your topic.

>>> Topic before supervisor

If you want to find a topic, and then a supervisor to suit it.

There are (at least) three approaches:

* Think of a question. Keep reading until you find the answer. Repeat until you find a question that nobody knows the answer to.

* Read in your chosen field until you come across a recent paper where you think ‘‘I could do that better.’’

* Think of a problem you have, for which you don’t have a solution.

You have found the beginning of your topic.

>>> Supervisor before topic

1. Recall your UG papers. Which did you most enjoy/find interesting? What about something different?

Once you know which field you want to focus on

2. Look through *all* the academic CS staff profiles, to find those who work in that field.
3. Be prepared for them to have nothing of interest.
4. Think of particular topic areas you are most keen on.
5. Email them asking if they have projects in that topic you could do for your dissertation.
6. Wait a while.
7. If they haven't responded after two weeks, email again.

>>> So you have found yourself a supervisor

1. Arrange an initial meeting at their convenience

2. Present your topic ideas to them; and/or

2. Listen to the topic ideas they have

3. Select your favourite, and refine it into a proposal
    * What is the motivation for it? (Why)
    * What do you hope to achieve? (What)
    * Will you need any extra equipment? (How)
    * Work at a desk or in your supervisor's lab? (Where)
    * Do you need to involve anyone else? (Who)
    * Can you estimate a timeline? (When)

4. Develop that into a plan with your supervisor

It is *never* too early to get started on this.

Section 2

700 Prerequisites

>>> Logically dependent

* Stated prerequisites are usually fairly logical

* If you have high enough grades otherwise, you *might* not need to meet them all

* Can make a concession request. Result up to enrolments staff

* Basically:
    * Do you have sufficient background knowledge?
    * Will you be able to keep up with this paper?

* If in doubt, contact the teaching staff for the paper

>>> General Preparations

Recommendations that will apply to just about every paper.

* CS 210, CS 220, CS 225 & CS 320 if possible

* All recommended and required preparation listed on a course's description

* Understanding of the basics of algorithms & data structures

* Strong knowledge of at least one programming language

* Familiarity with fundamental mathematics
    * Linear Algebra -- Vectors and matrices
    * Calculus -- Differentiation and integration
    * Discrete Mathematics -- Graph theory

* Willingness to learn about whatever is needed

>>> In particular

For every paper in X I recommend all of Y:

* CS 711, CS 732, CS 734
    1. CS 335
    2. At least one of CS 230, CS 235, CS 280
    3. CS 320, CS 351
    4. CS 340
    5. CS 350
    6. CS 313

>>> In particular

For every paper in X I recommend all of Y:

- CS 711, CS 732, CS 734
    1. CS 335
    2. At least one of CS 230, CS 235, CS 280
    3. CS 320, CS 351
    4. CS 340
    5. CS 350
    6. CS 313

- CS 715, CS 771, CS 773
    1. CS 373, Maths 208/250
    2. CS 369
    3. CS 320
    4. CS 313

>>> In particular

For every paper in X I recommend all of Y:

* CS 705, CS 747
    1. CS 345

>>> In particular

For every paper in X I recommend all of Y:

- CS 705, CS 747
    1. CS 345

- CS 702, 725
    1. CS 313, CS 340
    2. CS 316
    3. CS 314

>>> In particular

For every paper in X I recommend all of Y:

* CS 705, CS 747
    1. CS 345

* CS 702, 725
    1. CS 313, CS 340
    2. CS 316
    3. CS 314

* CS 726, CS 727, CS 742
    1. CS 314, CS 316
    2. For CS 727, stage 3 Maths

>>> In particular

For every paper in X I recommend all of Y:

* CS 705, CS 747
    1. CS 345

* CS 702, 725
    1. CS 313, CS 340
    2. CS 316
    3. CS 314

* CS 726, CS 727, CS 742
    1. CS 314, CS 316
    2. For CS 727, stage 3 Maths

* CS 720, 750
    1. CS 320, CS 350

>>> In particular

For every paper in X I recommend all of Y:

* CS 751, CS 752, CS 753
    1. CS 351
    2. CS 369
    3. CS 320

>>> In particular

For every paper in X I recommend all of Y:

* CS 751, CS 752, CS 753
    1. CS 351
    2. CS 369
    3. CS 320

* CS 760, CS 761, CS 765, CS 767
    1. CS 367
    2. CS 320, CS 350
    3. CS 369
    4. CS 351

Section 3

Some Myths About Postgrad CS

>>> Myths

‘‘PG CS is {}’’

* all maths

>>> Myths

‘‘PG CS is {}’’

* all maths

* _no_ maths

>>> Myths

‘‘PG CS is {}’’

- all maths

- *no* maths

- all theory

>>> Myths

‘‘PG CS is {}’’

- all maths
- *no* maths
- all theory
- *no* theory

>>> Myths

‘‘PG CS is {}’’

* all maths

* *no* maths

* all theory

* *no* theory

* all programming

>>> Myths

‘‘PG CS is {}’’

- all maths
- *no* maths
- all theory
- *no* theory
- all programming
- *no* programming

>>> Myths

‘‘PG CS is {}’’

* all maths

* *no* maths

* all theory

* *no* theory

* all programming

* *no* programming

* all writing

>>> Myths

‘‘PG CS is {}’’

* all maths
* *no* maths
* all theory
* *no* theory
* all programming
* *no* programming
* all writing
* *no* writing

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG

* PG CS won’t be much work

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG
* PG CS won’t be much work
* You have to be some sort of genius to do PG CS

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG

* PG CS won’t be much work

* You have to be some sort of genius to do PG CS

* You have to be an amazing programmer to do PG CS

>>> Myths 2: The Falsehoods

- PG CS will be easier than UG
- PG CS won’t be much work
- You have to be some sort of genius to do PG CS
- You have to be an amazing programmer to do PG CS
- You will be judged on the ‘quality’ of your code

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG

* PG CS won’t be much work

* You have to be some sort of genius to do PG CS

* You have to be an amazing programmer to do PG CS

* You will be judged on the ‘quality’ of your code

* There’s no benefit to good code quality in PG

>>> Myths 2: The Falsehoods

- PG CS will be easier than UG
- PG CS won’t be much work
- You have to be some sort of genius to do PG CS
- You have to be an amazing programmer to do PG CS
- You will be judged on the ‘quality’ of your code
- There’s no benefit to good code quality in PG
- I’m already an expert, there’s no point in PG CS

>>> Myths 2: The Falsehoods

* PG CS will be easier than UG

* PG CS won’t be much work

* You have to be some sort of genius to do PG CS

* You have to be an amazing programmer to do PG CS

* You will be judged on the ‘quality’ of your code

* There’s no benefit to good code quality in PG

* I’m already an expert, there’s no point in PG CS

* PG CS has no relevance to a professional developer

>>> Myths 2: The Falsehoods

- PG CS will be easier than UG
- PG CS won’t be much work
- You have to be some sort of genius to do PG CS
- You have to be an amazing programmer to do PG CS
- You will be judged on the ‘quality’ of your code
- There’s no benefit to good code quality in PG
- I’m already an expert, there’s no point in PG CS
- PG CS has no relevance to a professional developer
- Topics in PG CS have no relation to ‘day-to-day’ programming

>>> Episode 3: Revenge of the Myths

* You will always be able to learn everything you need to for a given subject from a single textbook

>>> Episode 3: Revenge of the Myths

- You will always be able to learn everything you need to for a given subject from a single textbook

- You can learn everything you need just by reading research papers

>>> Episode 3: Revenge of the Myths

* You will always be able to learn everything you need to for a given subject from a single textbook

* You can learn everything you need just by reading research papers

* There's no benefit to reading research papers, beyond meeting course requirements

>>> Episode 3: Revenge of the Myths

* You will always be able to learn everything you need to for a given subject from a single textbook

* You can learn everything you need just by reading research papers

* There's no benefit to reading research papers, beyond meeting course requirements

* You will always learn something the first time you read a research paper

>>> Episode 3: Revenge of the Myths

- You will always be able to learn everything you need to for a given subject from a single textbook

- You can learn everything you need just by reading research papers

- There's no benefit to reading research papers, beyond meeting course requirements

- You will always learn something the first time you read a research paper

- Research papers are always well-written so you understand them immediately

>>> Episode 3: Revenge of the Myths

* You will always be able to learn everything you need to for a given subject from a single textbook

* You can learn everything you need just by reading research papers

* There’s no benefit to reading research papers, beyond meeting course requirements

* You will always learn something the first time you read a research paper

* Research papers are always well-written so you understand them immediately

* There are never errors in research papers

>>> Episode 3: Revenge of the Myths

* You will always be able to learn everything you need to for a given subject from a single textbook

* You can learn everything you need just by reading research papers

* There's no benefit to reading research papers, beyond meeting course requirements

* You will always learn something the first time you read a research paper

* Research papers are always well-written so you understand them immediately

* There are never errors in research papers

* Papers from other CS topics are of no relevance to you

>>> Oh No! More Myths

- The final write-up always clearly reflects how the researcher arrived at the end result

>>> Oh No! More Myths

* The final write-up always clearly reflects how the researcher arrived at the end result

* Once you have chosen a subject area, you can only work in that area forever more

>>> Oh No! More Myths

* The final write-up always clearly reflects how the researcher arrived at the end result

* Once you have chosen a subject area, you can only work in that area forever more

* You'll never have to do a group project in PG CS

>>> Oh No! More Myths

- The final write-up always clearly reflects how the researcher arrived at the end result

- Once you have chosen a subject area, you can only work in that area forever more

- You'll never have to do a group project in PG CS

- You'll achieve everything you expect to in your dissertation

>>> Oh No! More Myths

* The final write-up always clearly reflects how the researcher arrived at the end result

* Once you have chosen a subject area, you can only work in that area forever more

* You’ll never have to do a group project in PG CS

* You’ll achieve everything you expect to in your dissertation

* Your dissertation will be completely novel

>>> Oh No! More Myths

- The final write-up always clearly reflects how the researcher arrived at the end result
- Once you have chosen a subject area, you can only work in that area forever more
- You’ll never have to do a group project in PG CS
- You’ll achieve everything you expect to in your dissertation
- Your dissertation will be completely novel
- Your dissertation *must* be completely novel

>>> Oh No! More Myths

* The final write-up always clearly reflects how the researcher arrived at the end result

* Once you have chosen a subject area, you can only work in that area forever more

* You'll never have to do a group project in PG CS

* You'll achieve everything you expect to in your dissertation

* Your dissertation will be completely novel

* Your dissertation *must* be completely novel

* Your supervisor will always be available to suit your schedule

>>> Oh No! More Myths

- The final write-up always clearly reflects how the researcher arrived at the end result
- Once you have chosen a subject area, you can only work in that area forever more
- You'll never have to do a group project in PG CS
- You'll achieve everything you expect to in your dissertation
- Your dissertation will be completely novel
- Your dissertation *must* be completely novel
- Your supervisor will always be available to suit your schedule
- Researchers always know exactly what they're doing

Section 4

Parting Words

Postgraduate Computer Science is a very valuable (and enjoyable) experience, but it is a lot of work and *definitely* isn't for everyone.

If you are here, however, it just might be for you.

Always remember

If you knew what you were doing, it probably wouldn't be research.