

A few tips I wish I had when I was an undergrad

컴퓨터 공학부 IT-리더십 세미나

2016년 6월 1일 오후 1시

카이스트 전산학부 교수 문수복



이 자료는 2016년 6월 1일 서울대 컴퓨터 공학부 2-4학년 학부생들을 대상으로 한 세미나 자료입니다.

4 Years of Knowledge for 40 Years of Prosperity?

Feasible at all? No.

대학 4년동안 배운 것으로 40년을 먹고 살 수는 없습니다. 대학에서는 40년 동안에도 계속 필요한 기초를 다지고, 그것을 기반으로 새로운 것을 “효과적”으로 배워나가는 방법을 체득해야합니다.

Learn *How to Learn* !

Where to Start?

■ Courses

- Textbooks
- Homework assignments and programming projects
- Exams

자, 전산학 전공을 시작하는 학생들이 누구나 가지고 있는 공통점이 과연 공부를 어디서 시작해야 할까 하는 질문입니다. 너무 빠른 대답 같지만 수업을 듣고, 교과서를 보고 공부하고, 숙제와 시험을 치루는 것입니다. 교과서가 어떻게 생겼나를 생각해보면 누군가가 많은 시간을 투자해서 전공 내용을 정리하고, 문제들도 만들고, 그것을 볼 사람들이 많다는 예상하에 출판까지 한 게 교과서입니다. 학부생용 교과서가 있다는 뜻은 해당 과목이 학부생들이 꼭 배워야 할만큼 중요하고 또한 그 내용이 정리되어 책으로 출판될만큼 학계에서는 어느 정도 교과과정에 대한 consensus가 생겼다는 뜻입니다. 반면 대학원에서 배울 과목들은 아직 교과서로 정리되지 못한 내용이 많아서 지금 나오고 있는 논문들 위주로 진행됩니다. 교과서가 있냐 없냐만 가지고도 해당 분야가 얼마나 자리잡힌 분야인지 아닌지를 알 수 있습니다.

세익스피어, 제인 오스틴, 헤밍웨이가 모두 정말 다른 스타일의 작가이듯이 교과서도 저자에 따라 굉장히 다른 스타일로 씌여집니다. 어떤 스타일의 교과서를 가지고 공부하겠냐는 학교에서는 교수가 정하지만, 나중에는 스스로 정해서 봐야하구요.

How Many Courses Should I Take?

- **Analytical Skills**

- Algorithm, Discrete Math, Linear Algebra, Statistics, Programming Languages (Automata)
- Complexity, Logic

- **Systems Skills**

- System Programming, Computer Architecture, Operating System, Compiler, Database, Network

- **Special Topics**

- AI/ML, Software Engineering, Graphics, HCI, Distributed Systems

2-3학년을 위한 전공 과목은 최대한 다 듣는게 좋습니다. 전산학에서도 어떤 연구를 해야할까, 나한테 맞는 적성이 뭘까 고민하면서 내가 이론스런 과목을 잘 하고 좋아하는지, 아니면 좀 시스템스러운 과목을 좋아하는지 정도만 알아채도 다음 단계 고민이 쉽지 않을까 생각합니다. 그래도 어찌 되었건, 위에 열거한 과목들은 꼭 듣고 나가야합니다.

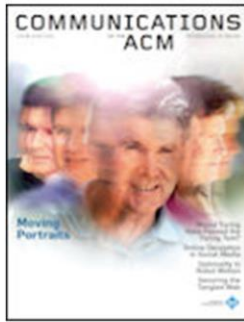
How to Connect Them All?

■ Capstone Project

- Goal: Synthesize all component knowledge about computer systems and build a new system as a team
- Plus: Improve communication and presentation skills

학부 때 OS에서부터 네트워크, 데이터베이스 등 전공수업을 다 듣고나서도 도 대체 컴퓨터 시스템이 어떻게 돌아가는지 잘 모를 수가 있습니다. 어찌 보면 각 컴포넌트 기술을 배우고 전체 시스템을 만드는 법을 배울 기회가 없었던 거죠. ACM Curriculum에서도 이를 위해서 Capstone Project를 추천합니다. 서너명이 팀으로 하나의 시스템을 설계부터 구현까지 다 하는 프로젝트를 해 보는 겁니다.

Research: Where to Begin?



교과과정 이외에서도 전공에 대한 정보를 얻을 수는 없을까요? 우선 WIRED 라는 잡지를 추천합니다. 트위터를 통해서 TechCrunch feed를 받아보는 것도 좋습니다. ACM의 Communications of ACM (CACM) 잡지는 최근 연구 동향에 대한 뉴스와 해당 분야 전문가들의 통시적 기고문, 그리고 맨 뒤에는 최근 학술대회에 발표된 논문들 중에서 특히 주목할 논문들을 해당 분야 중진의 해설과 같이 실습니다. 아무리 좋은 논문이라고 해도 그 분야 연구를 하지 않는 사람들은 왜 좋은지 잘 알기 어려울 수 있는데 해설이 같이 실려서 논문의 임팩트를 이해하는데 큰 도움이 됩니다. IEEE Spectrum 잡지는 IEEE 이라는 학회에서 학부생들을 대상으로 만든 잡지입니다. 공학 전분야에 걸친 내용을 다루지만 학부생 수준에 맞췄기 때문에 부담스럽지 않습니다.

Talks

| | Host | Target Audience | Quality |
|------------------|---------------------|---------------------|---------|
| Colloquium Talks | Professor in Charge | Grad Students | Medium |
| Invited Talks | Friend | Focus Group | Low |
| Job Talks | Future Employer | General CS Audience | High |

잡지 이외에 전공에 대한 공부를 할 수 있는 기회는 세미나입니다. 우선 학과에서 하는 콜로퀴엄이 있습니다. 이걸 대학원생을 대상으로 하지만 대학원생이라고 해도 특정 분야에 심화된 공부를 하는게 대부분이기 때문에 그렇지 않은 분야에 대해서는 학부 때 배운게 다 입니다. 콜로퀴엄에 모시는 분들은 연구를 활발히 하시는 어느 정도 명망있는 신진/중진 연구자분들이기 때문에 좋은 발표를 볼 확률이 높다. 발표 첫 5-10분 동안은 해당 분야 소개를 하고, 나머지 30-40분은 전공 얘기를 하는데, 처음 10-15분은 전산 공부를 하고 있는 사람이라면 학부생이라도 집중하면 알아들을 수 있게 개론적으로 하는데 통례이다.

그 다음에 교수가 개별적으로 초청해서 여는 세미나이다. 이걸 교수가 왜 초청을 했는냐에 따라 세미나의 질과 내용이 천차만별일 수 있다. 교수가 잘 아는 지인이 지나가는 길이길래 불러서 발표시키거나, 다른 학교에서 박사하고 있는 학생을 불러서 세미나시킬 수도 있고 등등. 아무리 그래도 모든 세미나의 처음 10-15분은 공짜 먹는 밥처럼 해당 분야 최근 소식을 듣는데 도움이 된다.

Job Talk는 대개 신입교원 지원자가 하는 세미나를 일컫는다. 이 세미나에는 지원자의 직장이 걸려있기 때문에 엄청나게 준비를 할 뿐만 아니라, 발표 맨

끝에는 앞으로 3-5년간 어떤 방향으로 연구를 할 것인지 포부도 밝혀야 하기 때문에 "좋은 발표"를 보고 배우려는 학생들에게는 적극 추천한다. 물론 발표를 다 잘한다는 보장은 없지만, 어찌 되었건 노력해서 준비한 세미나를 볼 기회가 된다.

모든 세미나의 처음 10-15분은 개론적 내용을 담기 때문에 학부생들에게도 큰 도움이 된다. 어떤 분야에 무슨 연구를 할까 고민하는 모든 학부생들에게 관련분야 세미나를 부지런히 들어보길 적극 권한다. 처음에는 못알아듣는게 대부분이겠지만 어느 정도 익숙해지면 누가 발표를 잘 하는지, 포인트를 명쾌하게 설명하는지, 쉽게 판가름할 수 있어진다. 그저 자꾸 들어 익숙해져야 할 뿐.

How to Find Key Papers to Read

- **"Google Scholar" search on keywords**
 - Identify seminal papers and surveys
- **Identify key players of the field**
 - Read thru their seminal papers
- **Review recent 3-5 years of publications of top conferences**
 - Browse the session titles

일단 Google Scholar에 가서 키워드로 검색되는 자료중에 서베이 논문이 있는지 찾아봅니다.

서베이 논문들은 지금까지의 연구를 잘 정리해놓기 때문에 개별 논문과 교과서 사이의 과정이라고 생각하면 됩니다.

헌데 서베이는 교과서를 쓸 마음이 있는 사람들이나 중진들이 쓰는데, 연구 업적으로 잘 인정되지 않아서

모든 분야에서 때맞춰 쓰여지지 않습니다. 해서 핵심 논문들을 찾아 읽어야 되기 쉽습니다.

근데 핵심 논문이 뭔가 찾는게 또 피인용횟수만으로는 힘듭니다.

이럴 때는 해당 분야 유명한 연구자 두세명의 연구 이력을 살펴보면 도움이 됩니다.

그리고 나서 해당 분야 최고 학회의 지난 몇년간 논문을 뒤져 봅니다.

헌데 학부생이 학회 논문을 한두편도 아니고, 수십-수백편을 읽어본다는 건 어렵습니다.

그쯤되면 박사받아서 나가거든요.

학술대회 세션이 있다가 없어졌는지, 새로 생긴 세션은 무엇인지 프로그램 구성만 살펴봐도

최근 연구 동향이 보입니다. 해서 프로그램을 살펴보고 싶 권합니다.

Read “Good” Writings



우리나라 토박이 학생들에게 아이비리그를 나온 원어민같은 수준을 바라면 안된다. 그래도 적어도 80프로 수준은 해줘야, 같이 작업하는 다른 사람들이 draft를 읽어도 이게 무슨 소리지 가우뚱을 덜하게 되리라는 것이다. 그리고 그 정도는 꾸준히 노력하면 된다. 그럼 어떻게 꾸준히?

영어를 잘하려면 “좋은 영어”를 많이 접해야 한다. 소설이 아니라 논픽션으로. 논문들도 좋지만, 사용되는 표현 등이 굉장히 제한되어 있고, 논리전개도 어찌 보면 뻔하기 때문에 신선함이 없다. 노는 날 방안에서 뒹굴며 손에 잡긴 쉽지 않다. 그럼 논문이 아닌 좋은 논픽션은 어디서 찾느냐? 매년 The New Yorker, The Atlantic Monthly, WIRED, Scientific American 등등 유수의 잡지에 실린 기사 중에서도 특히 명문장의 기사들을 선별해서 묶어내는 The Best American Science and Nature Writing를 강추한다. The Best Science Writing Online 2012를 살펴봤더랬는데 다 읽지는 못했지만, 아무래도 편집위의 검열을 거치지 않은 문장들이 많아서 조금 (아주 조금) 거칠다는 느낌을 받았다. 그래도 좋은 글이 많다. 이외에도 Best Travel Writing, Best Sports Writing 등의 시리즈가 있는데 다 좋다. 잡지 기사는 논문과는 달라서 인용하는 자료가 굳이 논문이 아니라 어떤 학파의 잘 알려진 가설이기도 하고, 다양한 분야에 알려진 통설이기도 하고. 그러나 논리적 전개만큼은 허술하지 않다. 그리고 전문적인 내용을 얼마나 쉽게 그리고 재밌게 써 내려가는지 읽어보면 큰

도움이 된다.

Programming Skill

- Introduction to Programming
- System Programming / OS / Compiler
- Programming Projects
- Open Source Project
 - “Measurable bias against women on GitHub”

프로그래밍은 꾸준히 하면 습니다.

40-50대가 되도 매일 꾸준히 몇 줄씩 코딩하겠다는 마음가짐으로~
하지만 오픈소스 커뮤니티에서조차도 여성들에 대한 ‘measurable bias’가 있다고 합니다.

그러니까 처음에 잘 못한다고, 남들이 알아주지 않는다고 기죽지 말고,
꾸준하게 천천히 내 템포대로 간다는 마음으로~

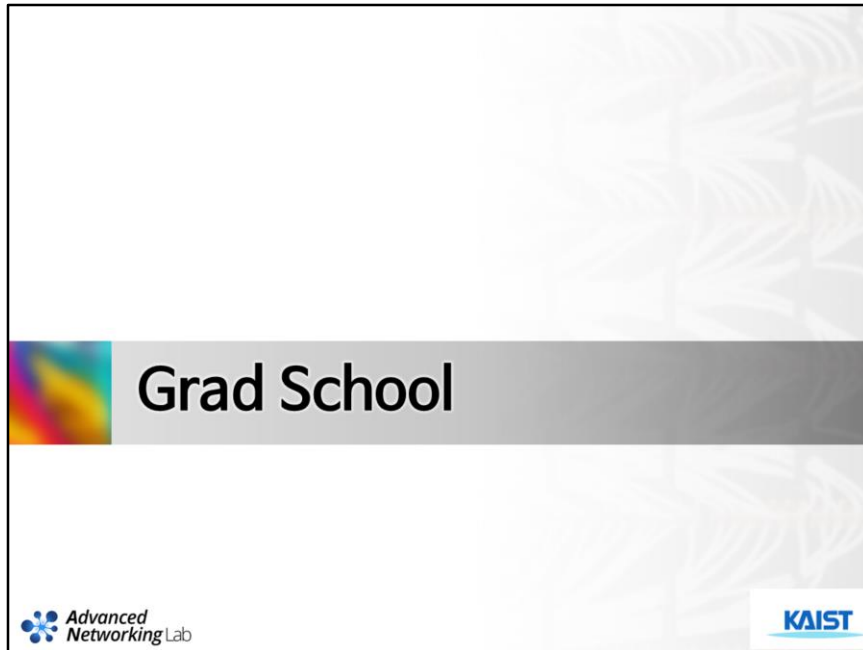
Read Good Books on Writing

- “Elements of Style” by W. Strunk Jr. and E.B. White
- “The Only Grammar Book You’ll Ever Need” by Susan Thurman

위의 두 책은 성문 종합영어++라고 생각하면 됩니다.
스터디 짜서 달달 외울정도로 공부하세요~

Your Value as a Member of Society

- Not proportional to your GPA
- Proportional to your service
 - Impact you made on other people's lives
 - Novel research output



대학원 진학할 때는 연구 분야도 중요하지만 지도교수와의 “rapport”도 중요하다. 지도교수한테 잘 맞추는 것도 요령껏 해야한다. 지도교수에게 무엇을 바래야할지, 어떻게 소통해야할지에 대해 얘기해보자. (꼭 지도교수와의 소통만이 아니다, 직장을 잡더라도 상사와의 소통에서도 어느 정도 적용될 것으로 생각된다.)

Challenge: Articulate Assumptions
You Make About Your Advisor and
Get Rid of Them

#A1 My advisor is omniscient (I)

- He probably knows
 - I could program decently
 - My spoken English is passable
 - I work most of the time (ahem!) *hard*
- But doesn't know / can't remember exactly
 - What languages I could program in
 - What papers I read last week and what I thought of them
 - Where I was stuck in my research the week before

Elevator pitch not only for a Job



Keep him/her up-to-date!
5분만 시간 있으세요?!!

#A2 My advisor is omniscient (II)

- S/he knows what research problem is for me
 - She has read and remembers my statement of purpose
 - She knows about funding initiatives
 - She has ideas she would like to work on
- But s/he doesn't know if I would actually want to work on it

Expose yourself as much as you can!

- Ask questions on all things technical but you don't know about or haven't heard about and answer them
- Try to answer questions your advisor asks
They ask because they don't have answers!!
- Get all colleagues around you involved in this Q&A dialogue

From my personal experience

- Help students pick up methodological approaches to good research
- Leave them with freedom on research topics

Then I go find funding after preliminary results...

#A3 My advisor is a superman!

- He would have read my email sent out at midnight and replied by 8am next morning
- He would work with me on my paper as hard as I
- He could write a recommendation letter in a day

But ...

- You are 1 of N students
- Your paper is 1 of M papers
- He reviews P papers a year
- He teaches Q courses a year
- He attends X committee meetings a week
- He travels Y times a year

Do the math!!! He only has $[(365 - 14 - 52 \times 2 - Y \times 5) \times (8 - X/5) - 4 \times P - 3 \times Q \times 16] / (N \times M \times 52)$ hours / week for you.

N=5, M=10, P=20, Q=2, X=5, Y=4 Answer =0.54

Then ...

- You realize how busy your advisor is
- You become so understanding that you don't have the heart to ask him for time

Think hard about what you want from him

- News on new research initiatives?
- Feedback on your research progress?
- Opinion on related work?
- Comments on your writing?

Turn every session to self-Q&A

- Instead of “What do you think about XXX?” ask “I think XXX should work because of A, B, and C in decreasing order of importance. So I plan to start with A. What do you think?”

Prioritize what he can do for you



- "I would appreciate feedback on the camera-ready version by this Friday"
- "Your schedule this week looks full, but can you squeeze in a half-hour meeting on Wed afternoon?"
- "The deadline for travel grant application is in two weeks. Thanks!"

#A4 My advisor could sub
as an English teacher



TOEFL GRE



Done



You are somewhere here



Some ideas in decreasing order of adoptability

- Give up on improving your spoken English in a few months
 - Instead, spend time on your slides
 - Script your talk
 - Every punchline you'd like to deliver, use pop-up text boxes

Q&A