COMP20230: Data Structures & Algorithms
Lecture 1: Introduction

Dr Andrew Hines

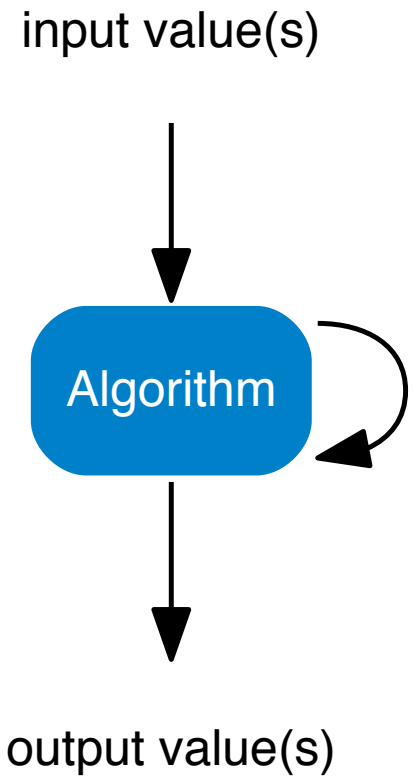
Office: E3.13 Science East
School of Computer Science
University College Dublin

andrew.hines@ucd.ie

Algorithm Definitions

Cormen et al., 2009 (Introduction to algorithms, 3rd ed.)

An algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output.



Algorithm Definitions

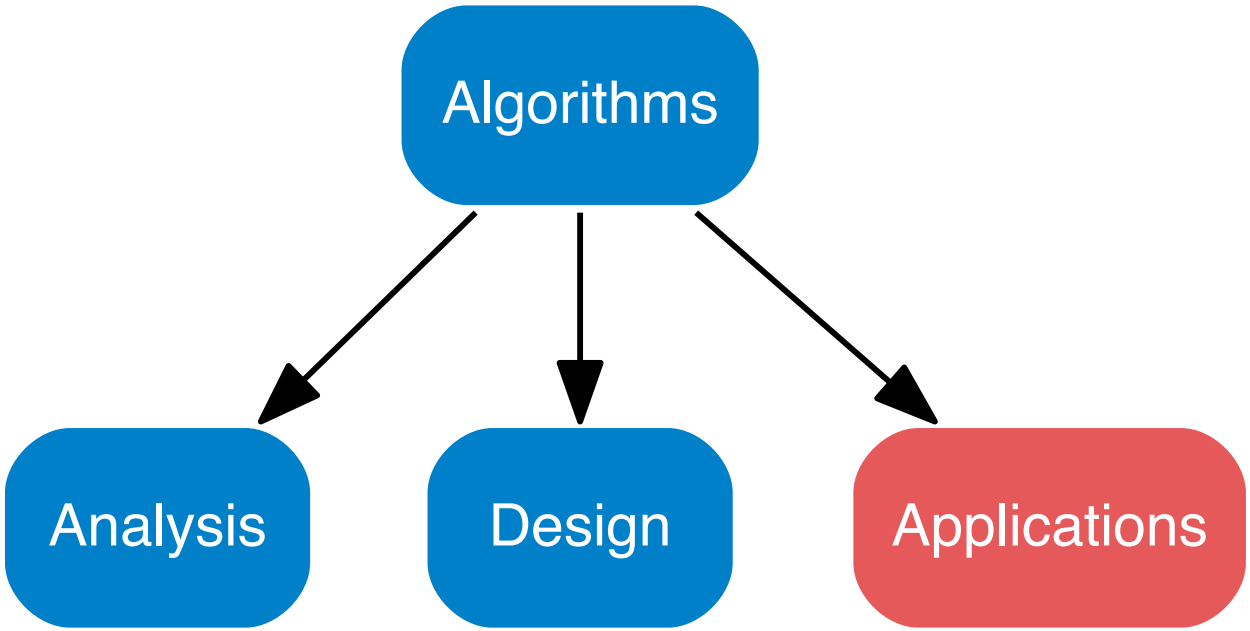
Boolos and Jeffrey (1974)

No human being can write fast enough, or long enough, or small enough[a] to list all members of an enumerably infinite set by writing out their names, one after another, in some notation. But humans can do something equally useful, in the case of certain enumerably infinite sets: They can give explicit instructions for determining the *n*th member of the set, for arbitrary finite n. Such instructions are to be given quite explicitly, in a form in which they could be followed by a computing machine, or by a human who is capable of carrying out only very elementary operations on symbols.

---

[a] "smaller and smaller without limit ...you'd be trying to write on molecules, on atoms, on electrons"

Algorithms



Scheduling and Optimisation Algorithms

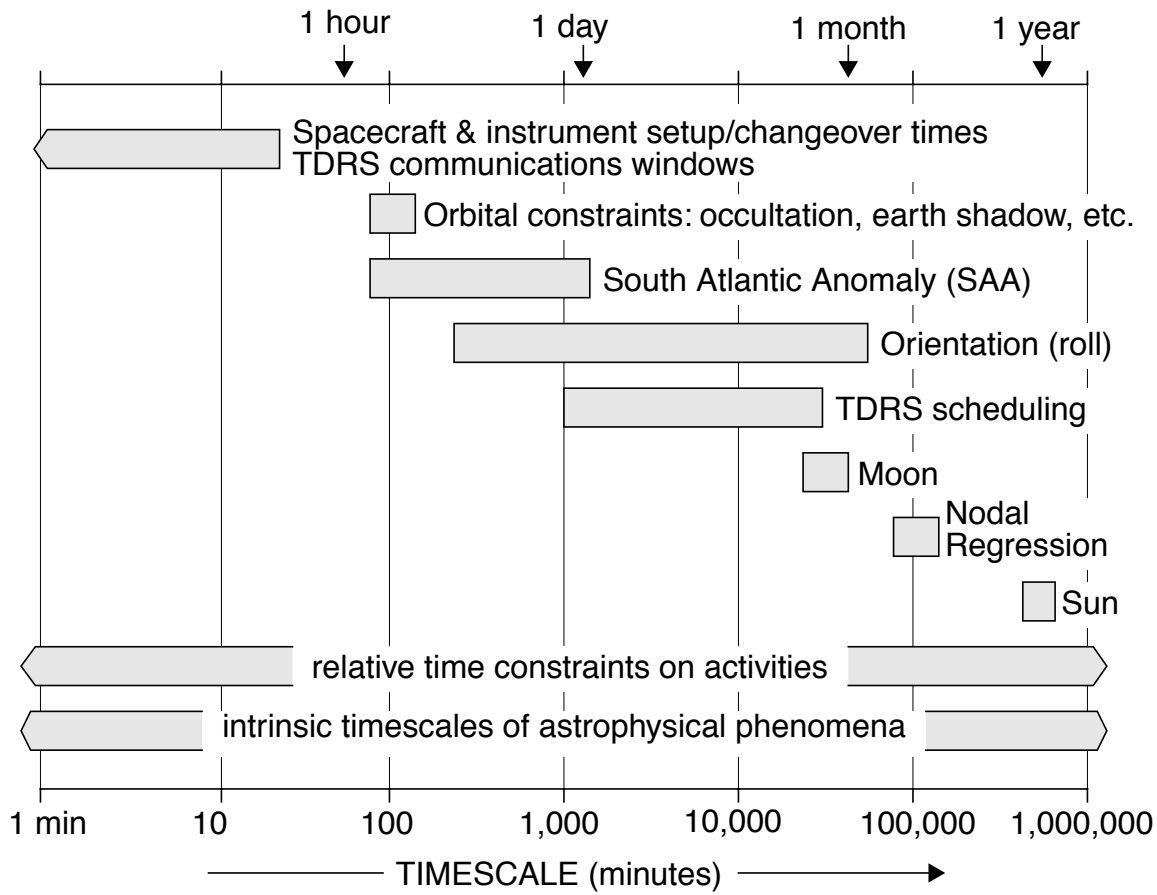
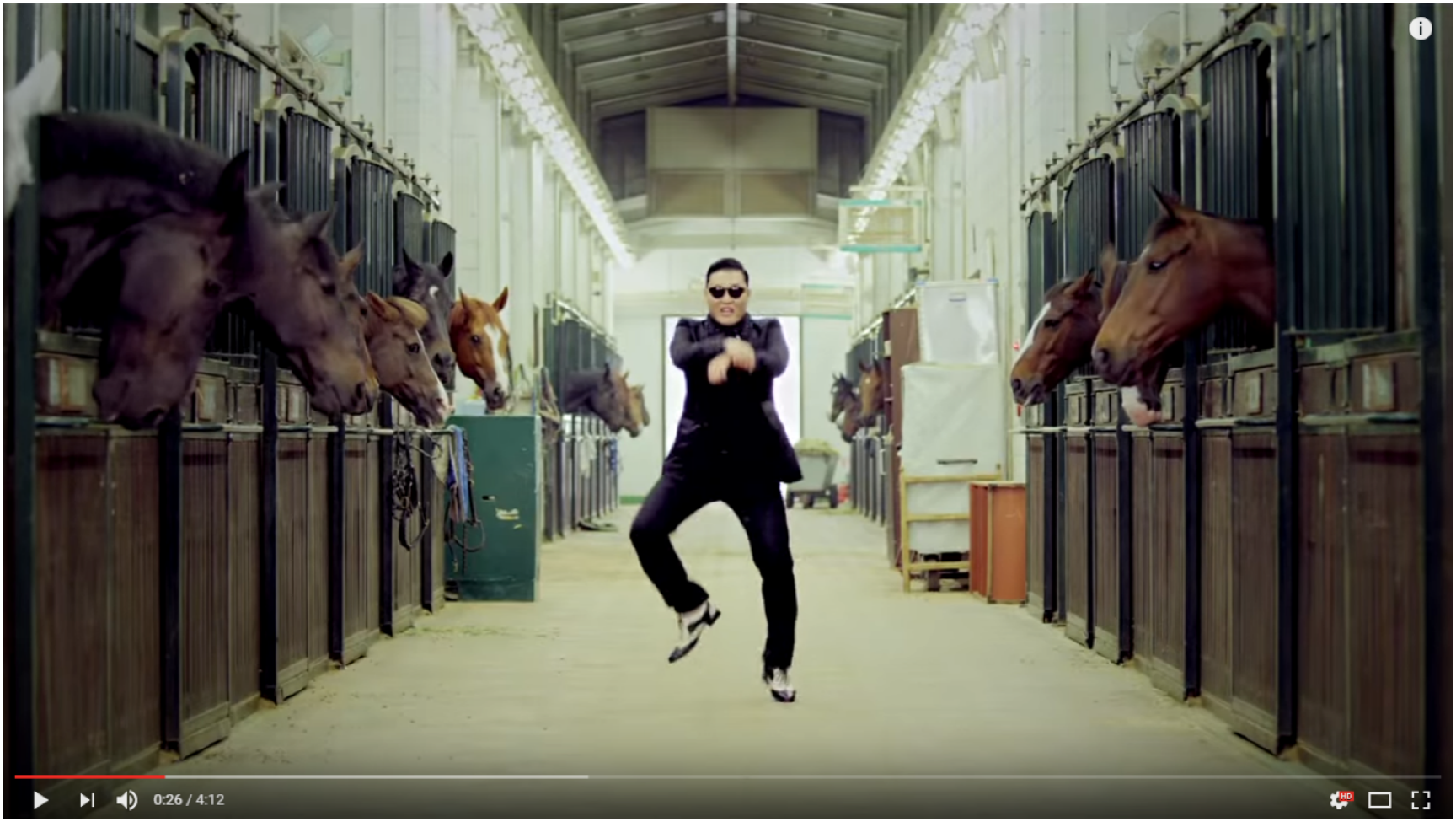
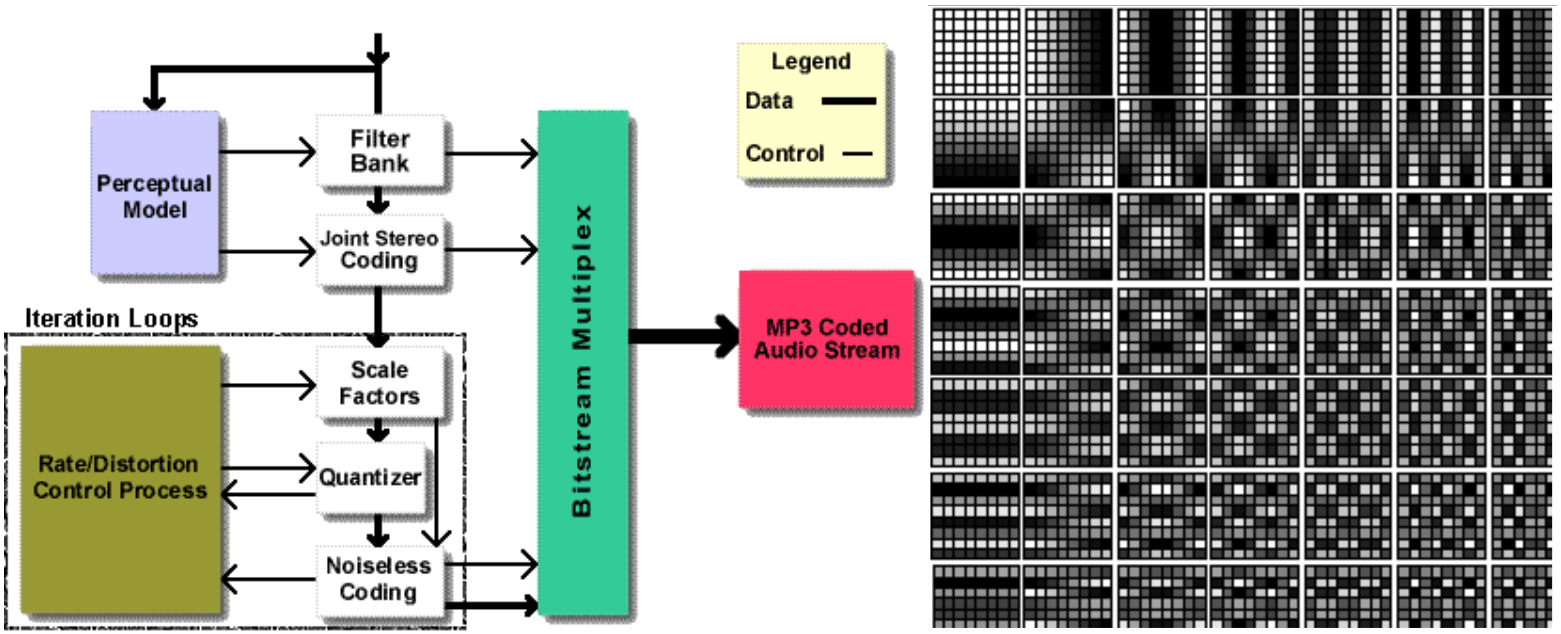
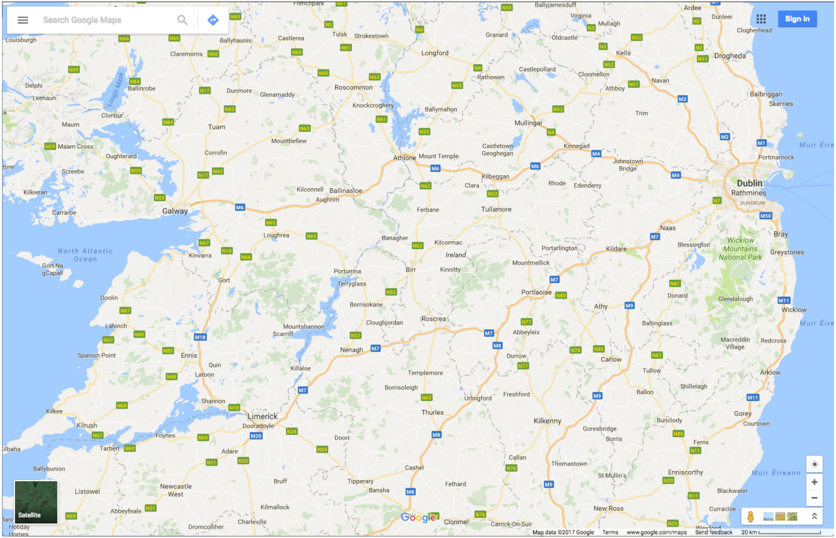


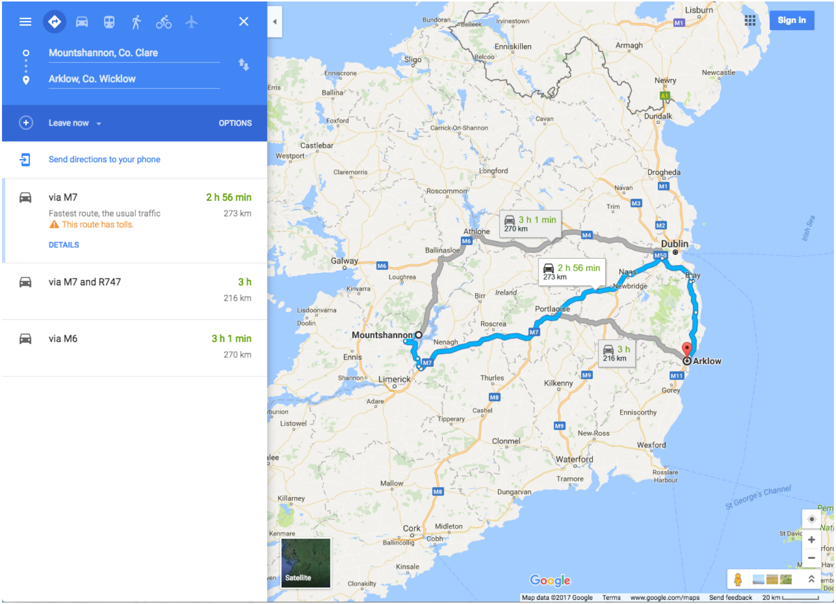
FIGURE 1. The range of timescales for HST scheduling constraints, covering more than six orders of magnitude.


0:26 / 4:12

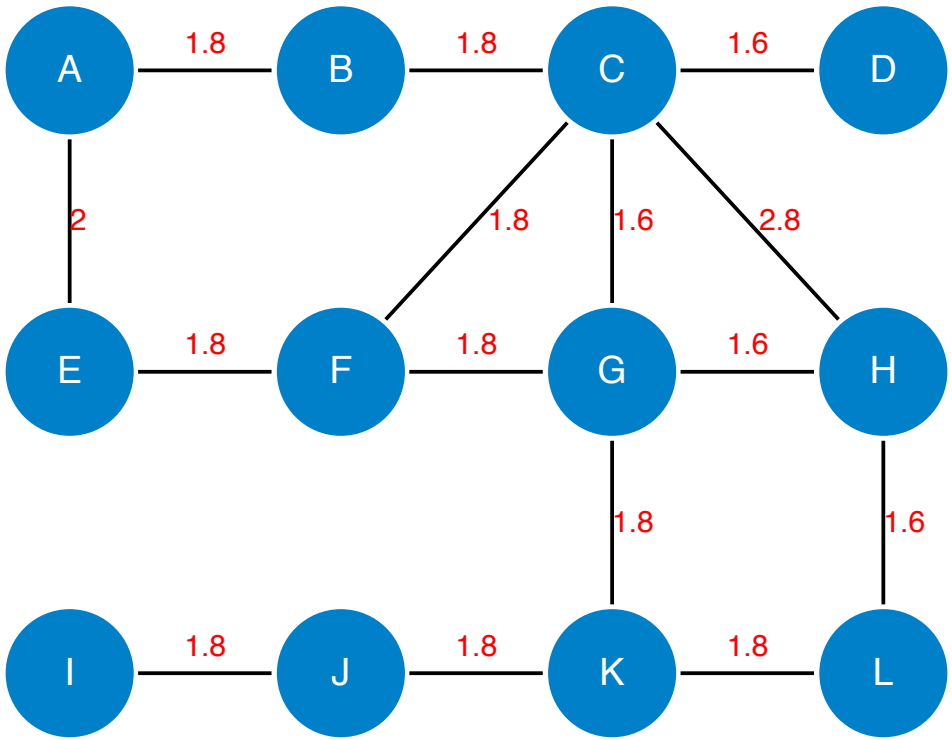
Audio and Video Compression Algorithms




Search Google Maps
Sign in
Louisburgh
Ballyhaunis
Castlerea
Tulsk
Strokestown
Granard
Virginia
Mullagh
Ardee
Dunleer
Clogherhead
Claremorris
Longford
Oldcastle
Kells
Drogheda
Castlepollard
Clonmellon
Rathowen
Delphi
Ballinrobe
Dunmore
Glenamaddy
Roscommon
Athboy
Navan
Leenaun
Lough Mask
Knockcroghery
Ballymahon
Balbriggan
Skerries
Clonbur
Maum
Tuam
Mullingar
Trim
Mountbellew
Maam Cross
Corrofin
Oughterard
Athlone
Mount Temple
Castletown Geoghegan
Kinnegad
Johnstown Bridge
Portmarnock
Muir Éireann
Screebe
Kilbeggan
Rhode
Edenderry
Dublin
Rathmines
Kilkieran
Kilconnell
Ballinasloe
Aughrim
Clara
Carraroe
Galway
Ferbane
Tullamore
Naas
Loughrea
Banagher
Kilcormac
Bray
Blessington
North Atlantic Ocean
Ireland
Portarlington
Kildare
Wicklow Mountains National Park
Greystones
Kinvarra
Gort Na gCapall
Portumna
Birr
Kinnitty
Mountmellick
Gort
Terryglass
Portlaoise
Donard
Glendalough
Doolin
Borrisokane
Athy
Wicklow
Cloughjordan
Roscrea
Baltinglass
Lahinch
Mountshannon
Scarriff
Abbeyleix
Macreddin Village
Redcross
Nenagh
Spanish Point
Ennis
Durrow
Carlow
Quin
Killaloe
Templemore
Tullow
Arklow
Latoon
Borrisoleigh
Shillelagh
Urlingford
Freshford
Ballon
Shannon
Kilkee
Thurles
Gorey
Limerick
Dooradoyle
Kilkenny
Bunclody
Courtown
Kilrush
Goresbridge
Foynes
Borris
Doon
Ferns
Tarbert
Adare
Callan
Thomastown
Enniscorthy
Ballybunion
Cashel
St Mullin's
Bruff
Tipperary
Fethard
Listowel
Newcastle West
Bansha
Blackwater
Kilmallock
New Ross
Curracloe
Abbeyfeale
Satellite
Charleville
Clonmel
Carrick-On-Suir
Holiday Homes
Dromcolliher
Google
Map data ©2017 Google
Terms
www.google.com/maps
Send feedback
20 km


Mountshannon, Co. Clare
Arklow, Co. Wicklow
Leave now
OPTIONS
Send directions to your phone
via M7
2 h 56 min
Fastest route, the usual traffic
273 km
This route has tolls.
DETAILS
via M7 and R747
3 h
216 km
via M6
3 h 1 min
270 km
Sign in
3 h 1 min
270 km
2 h 56 min
273 km
3 h
216 km
Mountshannon
Arklow
Dublin
Galway
Limerick
Cork
Waterford
Athlone
Portlaoise
Nenagh
Ireland
Satellite
Google
Map data ©2017 Google
Terms
www.google.com/maps
Send feedback
20 km

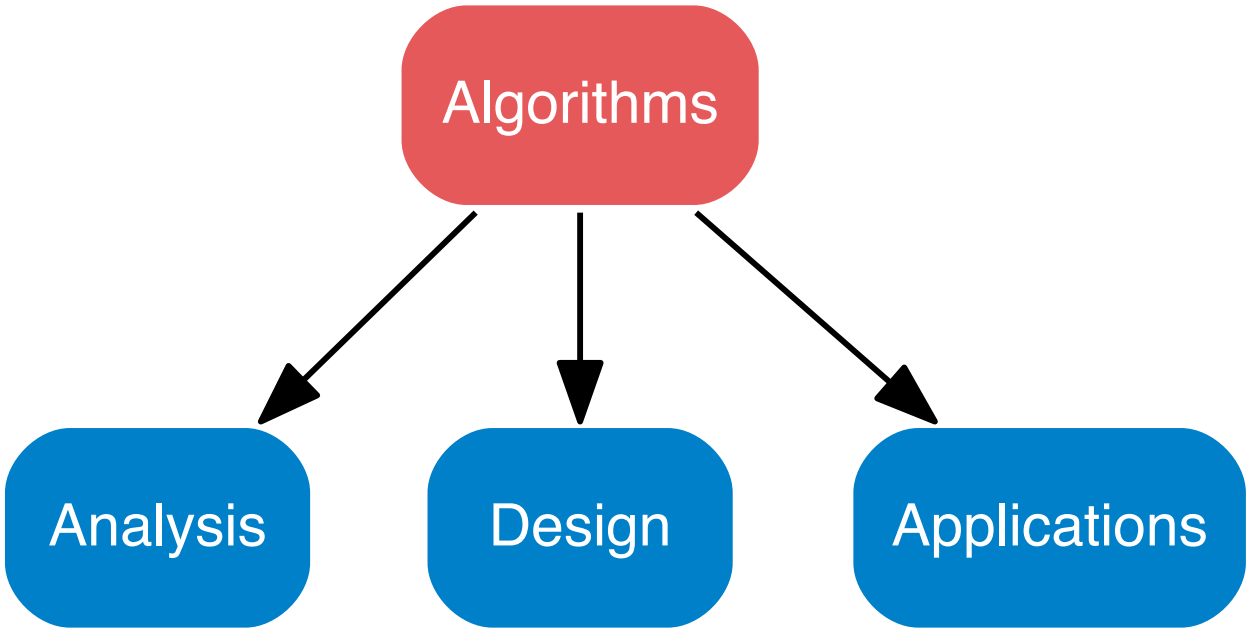
Routing Algorithms



Algorithms and Data Structures

Getting things done using a computer

Need: Algorithms and Data structures, language and paradigms
What: Methods and Analysis Algorithmic methods and doing it right (correctness)
How: Limitations and Robustness Cheaply (efficient), ability (can't do it == non computability?), Simplest (Occam's Razor)



Today

- Course Content - description, syllabus, learning outcomes
- Assessment and Credits - how much work is there to do?
- Why do it at all? - about me and you
- Structure of the module - Lectures and Labs
- Assignments and Study - Self directed learning

Module Description

This module provides the learner with the fundamental skills of design, development and assessment of algorithms and data structures.

Module Content

How we will deliver the Learning Outcomes?

Introduction
Algorithm analysis: What makes a good algorithm?
How to write code: Recursion
Storing data in structures: Array and Linked lists
Sequences, Stacks and Queues
Maps and Hash Tables
Sorting and Selection
Trees and Graphs
Using these data structures and algorithms to solve problems!

Moodle

Moodle Student Self Enroll

Module Code: COMP20230
Password : algo2019

http://csmoodle.ucd.ie/

Assessment

- Continuous Assessment is carried out throughout the semester
- Plagiarism will be monitored closely in all assignments
- All assessments will be submitted via the Moodle site:
  - http://csmoodle.ucd.ie/
- Additional course content will be posted on Moodle - check it regularly!

Study Time

- 1 semester = 30 ECTS credits
- 5 credits for each module
- 110 hours work

Activity	**Hours**
Practical	16
Autonomous Student Learning	58
Lectures	24
Tutorial	12
Total	110

Self study

3–5 hr/week

About Me

- Spent a decade in the software industry
  - Software product companies in the airline and financial sectors
  - Various roles from developer to software architect to Director of Software Development, to VP Engineering
- Spent time studying
  - MA, BAI Computer and Electronic Engineering (TCD, 1998)
  - MSc in Technology Management (NITM, UCD, 2003)
  - MBA (UCD, Smurfit Business School, 2004)
  - PhD in Electronic Engineering (Speech and Hearing Signal Processing, EE Dept, TCD, 2012)
  - PgDip 3rd Level teaching and learning (DIT, 2016)
- Lectured in TCD 2012–2014 and DIT 2014–2017 (e.g. Case Studies in Computing, OO Programming python, Speech and Audio Processing)
- Joined UCD academic faculty as Asst Prof. in 2017

About You

About You

Zero to Hero in Two Semesters

By the summer Group Project...

Need to be able to anaylse problem, design, develop, optimise, test, document

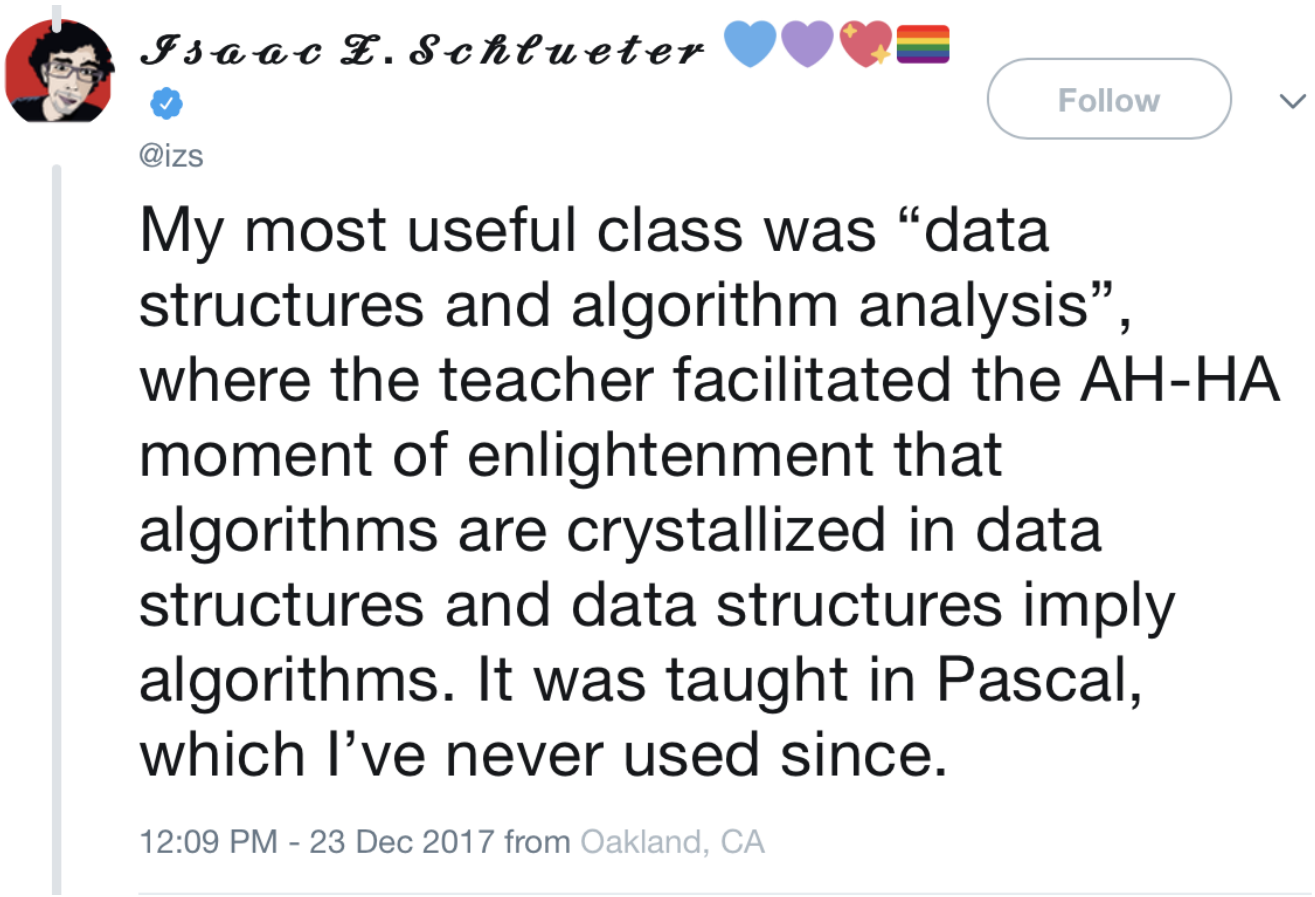
Apprentices

Learn through doing.

Practice, Practice, Practice

Isaac Z. Schlueter, npm[1] ceo

This module is useful[2] (and not just according to me):



[1] https://en.wikipedia.org/wiki/Npm_(software)

[2] https://twitter.com/izs/status/944661228705628160

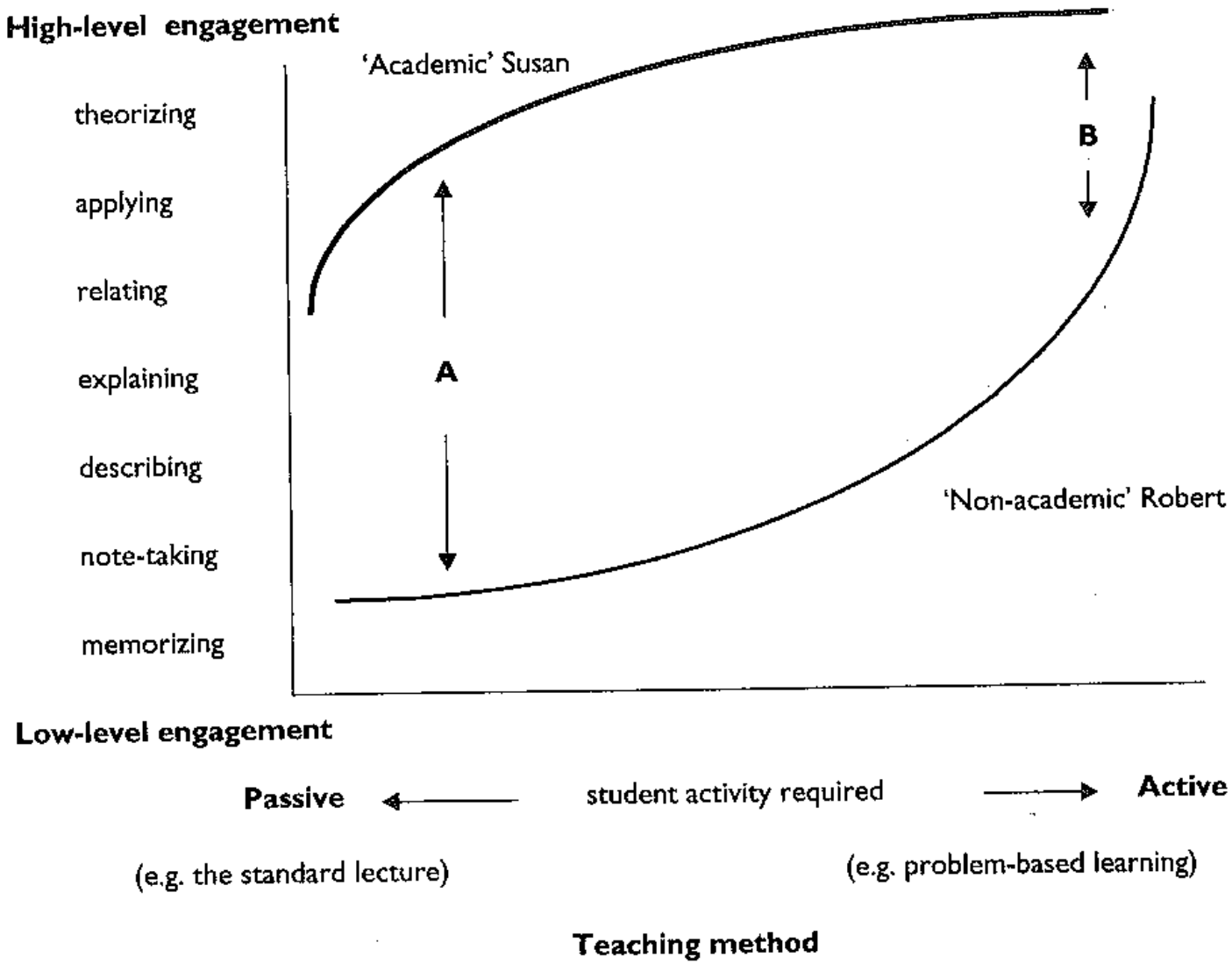
Linus Torvalds

"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships."

Linus Torvalds: principal developer of the Linux kernel at the core of Android, Linux, etc.

[1] https://en.wikipedia.org/wiki/Linus_Torvalds

How Do You Learn?



From: John Biggs, Teaching for Quality Learning at University, Open University Press, 2003

Programming is Hard, Algorithms and Data Structures are Harder

Typical experiences

Most work but lowest grade?!

I can't explain why it is hard but I know it is.

Analogy[1]

Studying poetry in a non-native language

Combines two parallel problems: I don't speak the language and I don't know much about poetry

Syntax: punctuation, word order: "colourless green ideas sleep furiously"; lexical categories (noun, verb, adjective, pronoun, determiner etc.)
Semantics: *logical semantics* (sense and reference and presupposition and implication) and *lexical semantics* (analysis of word meanings and their inter-relations)

[1] Inspired from the Practice of Computing Using Python, Punch, Enbody, Pearson Addison-Wesley 2011

French Poetry: Clement Marot (1496 – 1544)

Original and a literal translation by Douglas Hofstadter

"A une Damoyselle malade"

Ma mignonne
Je vous donne
Le bon jour;
Le séjuour
C'est prison.
Guérison
Recouvrez,
Puis ouvrez
Votre porte
Et qu'on sorte

My sweet/cute [one] (feminine)
I [to] you (respectful) give/bid/convey
The good day (i.e., a hello, i.e., greetings).
The stay/sojourn/visit (i.e., quarantine)
Itis prison.
Cure/recovery/healing (i.e., [good] health)
Recover (respectful imperative),
Andthen open (respectful imperative)
Your (respectful) door,
And [that one (i.e., you (respectful)) should go out

Douglas Hofstadter, Le Ton beau de Marot (1997)

French Poetry: Clement Marot (1496 – 1544)

Original and translation by Steven D. Jamar

"A une Damoyselle malade"

Ma mignonne
Je vous donne
Le bon jour;
Le séjuour
C'est prison.
Guérison
Recouvrez,
Puis ouvrez
Votre porte
Et qu'on sorte

My sweet dish,
You I wish
A good day.
Where you stay,
Is a jail.
Though so pale,
Leave your bed,
Regain red.
Ope' your door
Stay not, poor

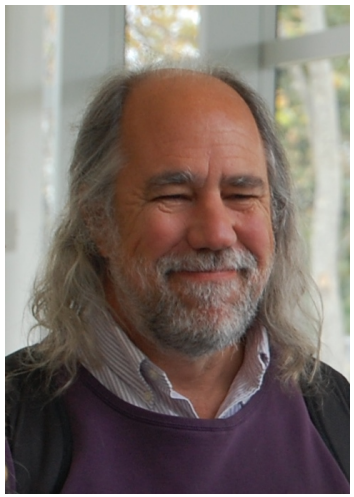
Programming, Syntax and Semantics

- *Syntax* of a particular programming language
- Reserved words, declaration of variables, functions, methods
- Style guidelines (e.g. PEP8 PEP20 >>> `import this`)
- Debug – interpreting error messages and stack traces
- Problem Solving and how to communicate the solution to a computer through a language

Good programming: focus of computer science

- How to let computers have our thoughts?
- Translating our "simple" thoughts and ideas into runnable code requires accuracy and rigor

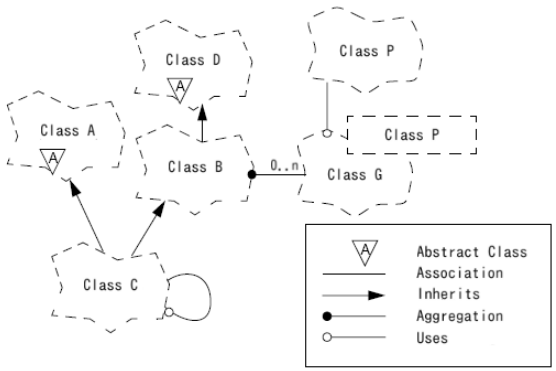
Programming, Syntax and Semantics



Grady Booch's fundamentals:

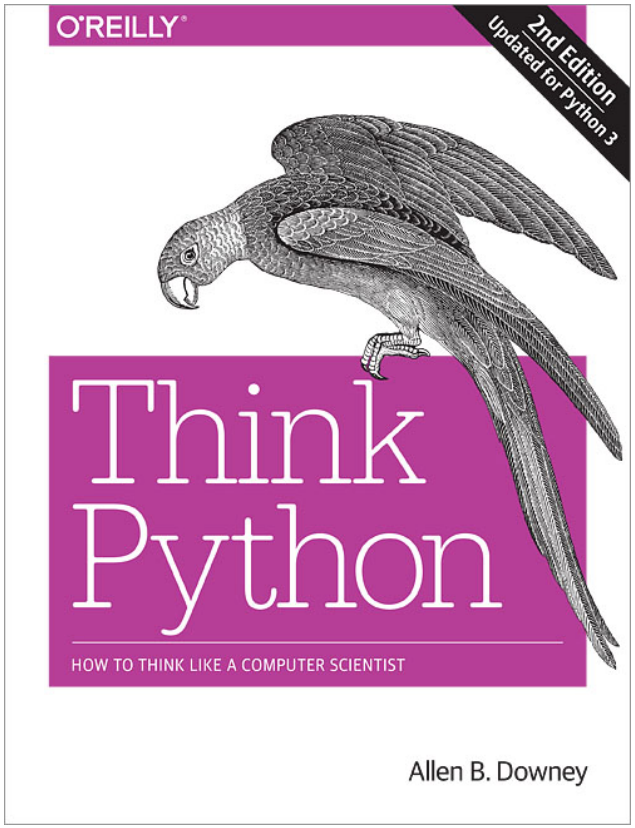
1. Create crisp and resilient abstractions.
2. Maintain a good separation of concerns.
3. Create a balanced distribution of responsibilities.
4. Focus on simplicity.
5. OOP - focus on the things rather than the algorithm process

The goal of a program is not to run, but to be read by BOTH computer and person.



Booch invented Universal Markup Language not as a replacement for readable code or documentation but to help communicate OOP relationships.

Free Python Textbook

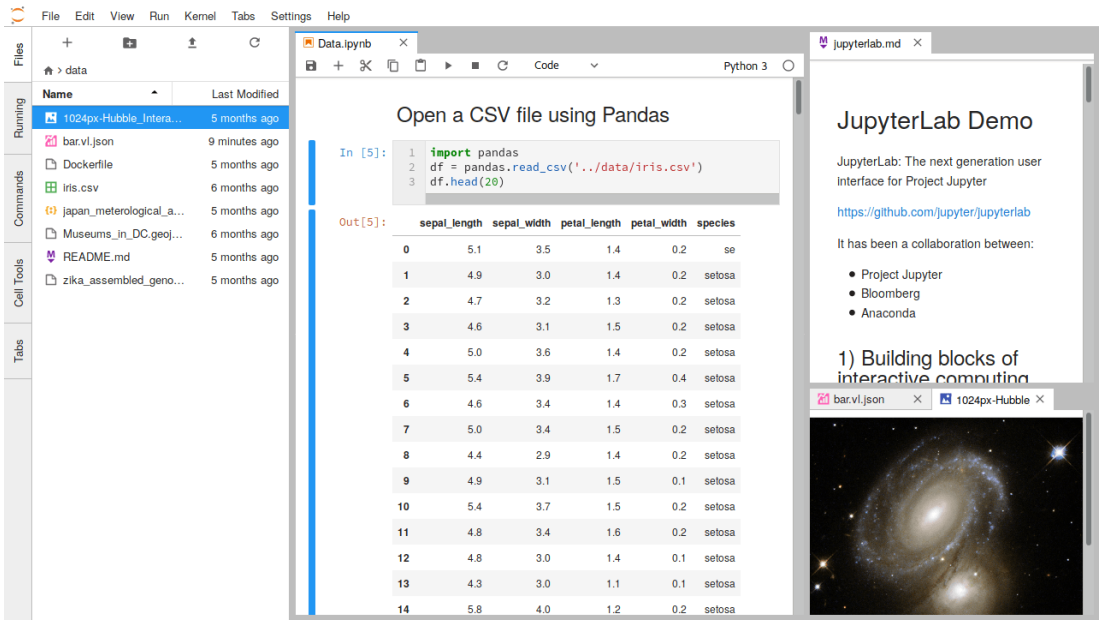


Think Python (2e).

Allen Downey

PDF on moodle

Coding Environment



JupyterLab

JupyterLab using Python 3 + anaconda.

`https://jupyterlab.readthedocs.io/en/latest/getting_started/overview.html#overview`

Learning Outcomes

On successful completion of this module the learner will be able to:

1. Understand how to determine the amount of resources (such as time and storage) necessary to execute a particular algorithm (algorithm analysis).
2. Understand the structure, nature and use of fundamental data structures including, Arrays, Linked Lists, Stacks, Queues, Trees and Dictionaries.
3. Implement the data structures in Python.
4. Understand the object-oriented programming constructs needed to encode a data structure and its access algorithms.
5. Design programs using these constructs to solve large problems.
6. Successfully write, compile, debug and run programs using these constructs.

Assessment

- Continuous Assessment: 60% (Throughout semester)
  - Mid-Term Test (15%)
  - Lecture Notes Preparation and Reflection Session (15%)
  - Group Project (30%)
- Examination: 40% (2 hr, end of semester exam, closed book)

School of Computer Science Grading

`https://www.cs.ucd.ie/Grading/`

Plagiarism

School and University plagiarism policies

Policies should be **read and understood** by all students.
There are **serious** consequences for confirmed instances of plagiarism.

School Policy

`https://www.cs.ucd.ie/sites/default/files/cs-plagiarism-policy_august2017.pdf`

The school policies and procedures should be read in conjunction with the UCD policy on Plagiarism and Academic Integrity[1] as well as the relevant sections (especially 6.2 and 6.3) of the UCD Student Code[2].

[1] `http://www.ucd.ie/registry/academicsecretariat/docs/plagiarism_po.pdf`
[2] `http://www.ucd.ie/registry/academicsecretariat/docs/student_code.pdf`

Plagiarism

ZERO TOLERANCE

- Plagiarism is a serious academic offence
- Our staff and demonstrators are proactive in looking for possible plagiarism in all submitted work
- Suspected plagiarism is reported to the CS Plagiarism subcommittee for investigation
  - Usually includes an interview with student(s) involved
  - 1st offence: usually 0 or NG in the affected components
  - 2nd offence: referred to the University disciplinary committee

Contact Details

andrew.hines@ucd.ie

- The best way to contact me is by email
- Feel free to get in touch with questions
- Don't expect immediate responses, I will endeavour to respond within 48 hours
- When emailing please state:
  - Your name (as it appears on Moodle),
  - Your class, and
  - Your student number.
- Use the forum, if you have a question you think the class would like to know the answer

Summary

Introduction covered

- Schedule, moodle, about me and you
- Why study algorithms and data structures – why is it hard?
- Text books, dev environment, Plagiarism
- Contact Details

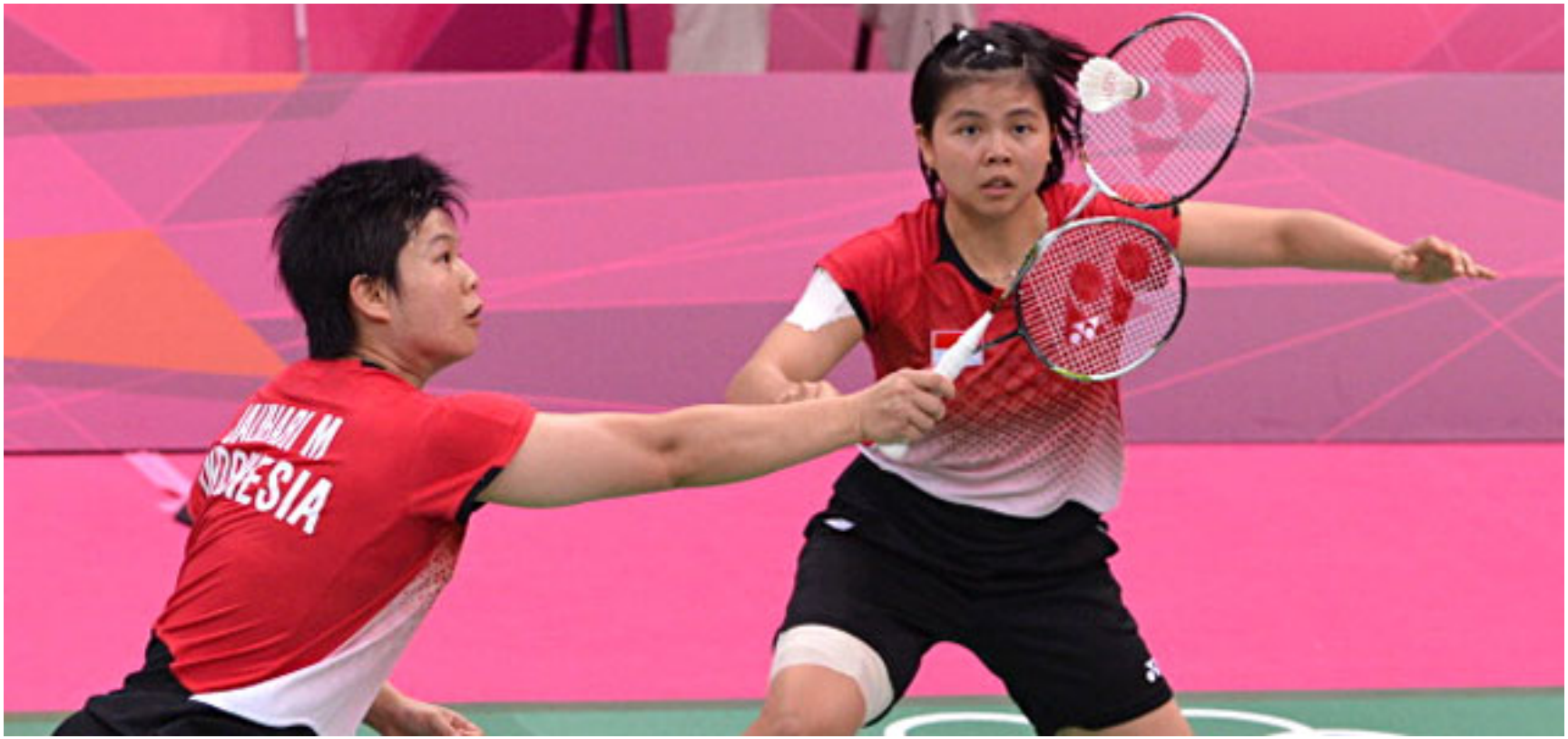
Tutorial

No tutorial session today

Tomorrow

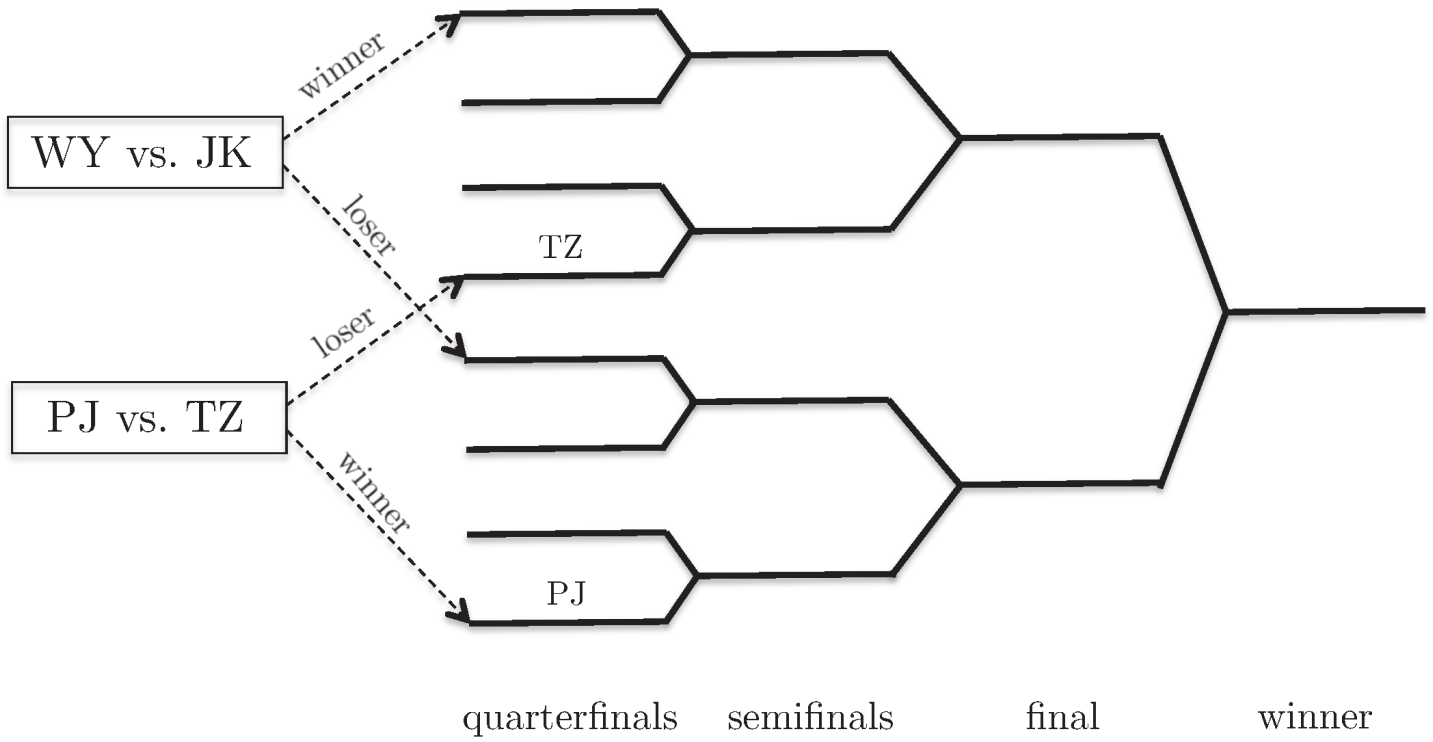
Algorithms and writing pseudocode, JupyterLab setup and familiarisation in practical session

Closing Tale: Badminton at the London 2012 Olympics



Women's Doubles Badminton at the London 2012 Olympics

Knockout format is a tree data structure. Algorithm decides which side of the tree.



Both teams, WY and JK preferred to play TZ in as late a round as possible.
8 women disqualified and no videos allowed on the web by the IOC

Tim Roughgarden, Twenty Lectures on Algorithmic Game Theory, Cambridge University Press, 2016