COMP20230: Data Structures & Algorithms Lecture 20: Module Review

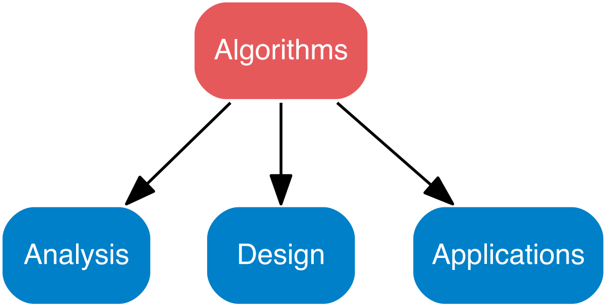
Dr Andrew Hines

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

andrew.hines@ucd.ie

Module Description

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



Learning Outcomes (What did we set out to achieve?)

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.

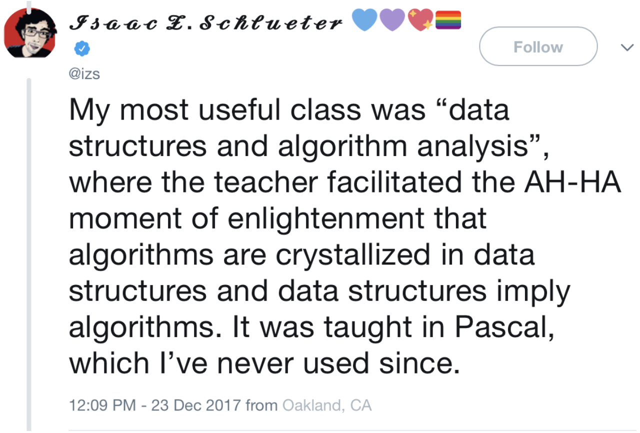
Module Content (What did we look at to achieve the LOs?)

Topic Areas

Complexity Analysis
Recursion
Abstract Data Types
Arrays and Linked Lists
Stacks, Queues and Circular Arrays
OOP and Testing
Sorting (Bubble, Merge, Insertion)
Trees, DFS, BFS
Graphs, Dijkstra
Minimum Spanning Trees, Prim and Kruskal
Binary Search Trees
Hash tables

Isaac Z. Schlueter, npm[1] ceo

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


Isaac Z. Schlueter 💙💜💖🏳️‍🌈

@izs

Follow

My most useful class was "data structures and algorithm analysis", where the teacher facilitated the AH-HA moment of enlightenment that algorithms are crystallized in data structures and data structures imply algorithms. It was taught in Pascal, which I've never used since.

12:09 PM - 23 Dec 2017 from Oakland, CA

1 https://en.wikipedia.org/wiki/Npm_(software)
2 https://twitter.com/izs/status/944661228705628160

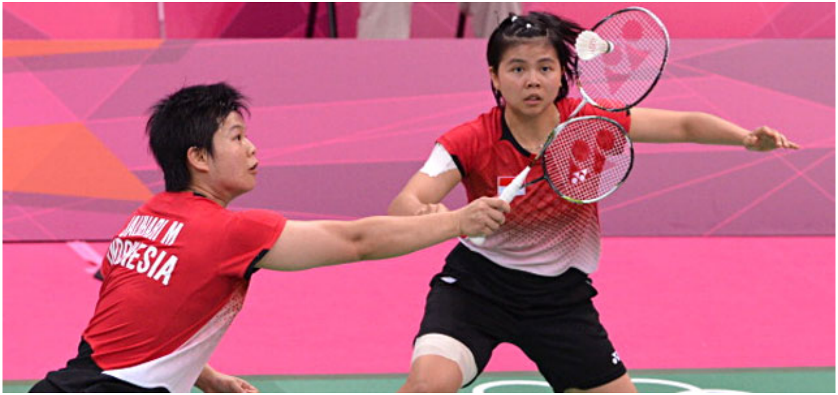
Linus Torvalds

"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.

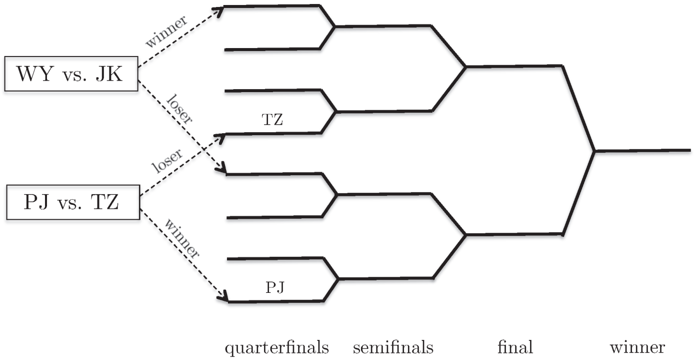
[1] https://en.wikipedia.org/wiki/Linus_Torvalds

The (Opening) 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

Data Structures

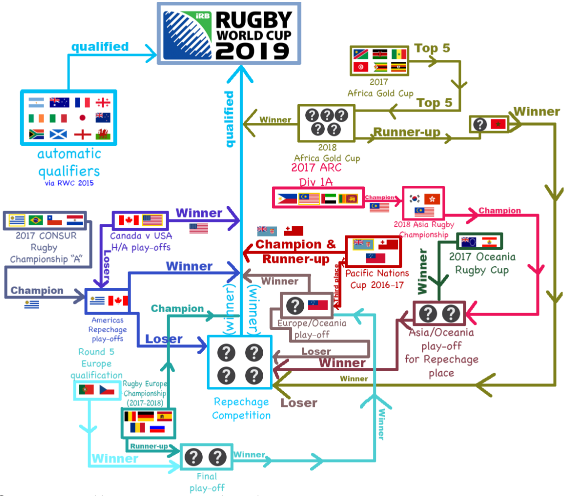
A tree for the knockout competition and graph to model the decisions and costs of moving states.

Algorithm

Goal mis-alignment between tournament organisers (want people to play to the best of their ability in all games). Teams (want to get best medal possible).

Rugby World Cup 2019

Limited and heterogeneous data can lead to complex algorithms!



Source: `https://en.wikipedia.org/wiki/File:Rugby_World_Cup_2019_Qualification_illustrated.png`

Attributes of a good algorithm

- Correctness
- Speed
- Efficiency
- Security
- Robustness
- Clarity
- Maintainability

RONSEAL
Does exactly what it says on the tin
5 YEAR
WOOD STAIN
WATERPROOFS & PROTECTS ALL EXTERIOR JOINERY
SATIN
NATURAL OAK
✓ 5 years guaranteed ✓ Doors & windows
✓ Rainproof in 1 hour ✓ Water-based formula

Trade-offs

- For the data (see last slide!)
- Between the algorithm and data structure
- Between attributes (e.g. maintainability vs. speed – amount of debug in logs)
- Between performance priorities (e.g. create, amend, delete, search)

RONSEAL
Does exactly what it says on the tin
5 YEAR
WOOD STAIN
WATERPROOFS & PROTECTS ALL EXTERIOR JOINERY
SATIN
NATURAL OAK
✓ 5 years guaranteed ✓ Doors & windows
✓ Rainproof in 1 hour ✓ Water-based formula

Skills

Analysis

Design

Do I need the same skills and knowledge for both?

Knowledge

- Physical Data Structures: Linked Lists and Arrays

Knowledge

- Physical Data Structures: Linked Lists and Arrays
- Iteration and Recursion (and re-factoring)

Knowledge

- Physical Data Structures: Linked Lists and Arrays
- Iteration and Recursion (and re-factoring)
- Algorithms: Searching, Sorting, Selecting, Scheduling, Routing

Knowledge

- Physical Data Structures: Linked Lists and Arrays
- Iteration and Recursion (and re-factoring)
- Algorithms: Searching, Sorting, Selecting, Scheduling, Routing
- Abstract Data Types: Sequences, Sets, Stacks, Queues, Trees, Graphs, Maps

What else did you learn?

Beyond the learning outcomes

Did you learn anything unexpected?

Other knowledge examples?

- A little bit of LaTEX?
- A little more Python?
- Knowing the problem without the data is not enough
- Knowing the data without the problem is not enough
- If your bookshelf is sorted by purchase date or colour you will find it hard to find books but easy to appreciate data structures and searching algorithms
- Syntax and structure of languages can constrain your problem solving (pseudo code frees the mind!)
- A lot of problems have elegant solutions that have already been designed – knowing which one to use takes time and experience
- Complexity and running time can be difficult to estimate as solutions grow bigger and more complex and we introduce recursion, complex data structures etc.

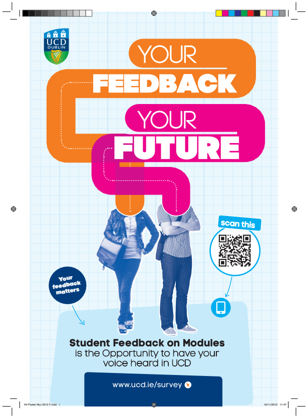
Other skills?

- Did reflect on the group project?
- Did you use GitHub (or wish you had?)
- Did you find that the teamwork helped or hindered you?
- Did you try to explain something to your teammates only to realise you didn't actually understand it yourself?
- Did they come up with very different but equally valid solutions to things?
- Did you learn to value clarity as an attribute?
- Did you find it hard to communicate technical information
- Did you make trade-offs?

What did you think of the module?

Module Survey

Assignments/assessment: Was there too much? What did you learn the most from?
Are there things you would have prefered to have skipped? Lecture notes, mid term test?
Structure of the lectures/labs/tutorials – How might you have got more out of them?
Would you like a more classroom style tutorial?



Take a few mins to provide feedback

Please: be constructive: we are human too (honest!)
Lecturers put a lot of time into preparation but we make mistakes

What did you think of the module?

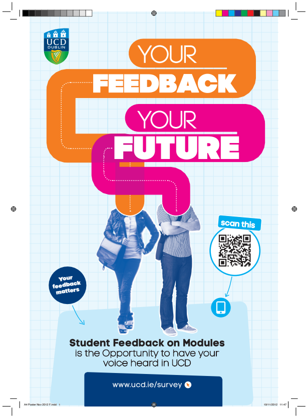
Module Survey

Please:

Take a few minutes to fill in the survey

What worked? Suggestions for improvement

`www.ucd.ie/survey`



Teaching and Learning Awards

If you particularly liked something

UCD has awards to acknowledge high quality teaching, learning, assessment and curriculum design.
If there was something (e.g. a novel assessment, teaching method) or someone who made a big difference to your experience over the year, consider nominating them for a T&L award.

...or just say thanks!

It's good to know if things worked as much as if they didn't.

The lecture notes (15%)

Lecture notes will be read and graded. Participation in the group session on Wednesday is also mandatory (alternative is to submit a written refection on 4 lecture notes other than your own).

Assessment Criteria:

1. **Clarity**: how well written, clear, and concise are they? How well do they explain the topic? How well are they formatted and presented?
2. **Completeness**: Do they cover the lecture topic? Do they go beyond the lecture to explain or give alternative examples to help understand?
3. **Correctness**: Are they technically correct? Have they errors or have things been misunderstood.

Lecture Notes sessions (15%)

Lab on Tuesday

Co-ordinate your reading and prepare for Wednesday (see below).

Assessment Criteria:

1. Each table should have circa. 8 people.
2. Each person gets 8 minutes presentation (+2 mins questions) to review 2 lectures (using the student lecture notes as input)
3. The groups should appoint a timekeeper and a chair to moderate the session
4. You can allocate the notes anyway you like but you cannot not review your own notes (!) and no set of notes should be reviewed twice.
5. Use the session today to design the best schedule to suit your group.

Lecture Notes sessions (15%)

Lab on Wednesday

Presentations: There will be a sign-in sheet and attendance and participation are worth 6% (the other 9% are for your lecture notes).

Question

Exam session tomorrow or next week?