CSCM12: software concept and efficiency
Introduction lecture

Cécilia Pradic
Swansea University, 30/01/2025

Who?

- Cécilia Pradic (she/her), lecturer
- Research topics: theoretical CS, logic, automata, computability theory
- **E-mail:** `c.pradic@swansea.ac.uk`
- **Personal webpage:** `https://cpradic.web.deuxfleurs.fr`
- **Background:**
  - PhD in Lyon/Warsaw, then postdoc Oxford; I've then been here for three years as a permanent.
  - I taught CSCM41J when I started, then CSCM12 each year. I teach functional programming (CS-205) in undergrad.

There used to be picture of me here because I was a conformist, but I don't see the point - if there is one, do let me know. Should you need to see what I look like, just come to the lectures.

What are we going to do?

- **Introduction** to algorithmics and datastructure
- Pre-requesites: CSCM41, basic math

Aims

By the end of the module, you should be able to:

- know how to compute the time complexity of your code
- know some standard algorithms and datastructures
- be able to solve simple algorithmic problems on your own
- and apply that knowledge to your programming!

Should be clear to you by the end of the module

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable,
Serializable
```

Resizable-array implementation of the `List` interface. Implements all optional list operations, and permits all elements, including `null`. In addition to implementing the `List` interface, this class provides methods to manipulate the size of the array that is used internally to store the list. (This class is roughly equivalent to `Vector`, except that it is unsynchronized.)

The `size`, `isEmpty`, `get`, `set`, `iterator`, and `listIterator` operations run in constant time. The `add` operation runs in *amortized constant time*, that is, adding n elements requires O(n) time. All of the other operations run in linear time (roughly speaking). The constant factor is low compared to that for the `LinkedList` implementation.

Is this a programming module?

In short, **no but yes**

- we will use java for motivation, examples and assessment
- but the focus won't be on learning java

But how do I benefit as a programmer?

- train to abstract away
- you will be able to write better more efficient code
- understand standard libraries and efficiency concerns

Will we need to do math?

In short, **yes a little bit**

- I will not focus on math/proofs
- But we will need to do some computations

(can't help it if we want to compute complexity)

Some of the things you shall be able to parse

$$n\log(n) + mn^2 = \mathcal{O}(mn^2)$$

$$f(n) = f\left(\lceil \tfrac{n}{2} \rceil\right) + K \qquad \text{implies} \qquad f(n) = \mathcal{O}(\log_2(n))$$

Further maths support

- some resources on Gary's slides from previous iterations I will put up
- Swansea's center for academic success **might** have resources
  e.g. `https://myuni.swansea.ac.uk/academic-success/maths-and-stats/`

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?
- limits (e.g. $\lim_{n \to +\infty} \frac{n^2}{2}$)

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?
- limits (e.g. $\lim_{n \to +\infty} \frac{n^2}{2}$)
- asymptotic notations (e.g., $\mathcal{O}(n \log(n) e^n)$ or $o(n^5)$)

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?
- limits (e.g. $\lim_{n\to+\infty} \frac{n^2}{2}$)
- asymptotic notations (e.g., $\mathcal{O}(n \log(n) e^n)$ or $o(n^5)$)
- proofs by induction

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?
- limits (e.g. $\lim_{n \to +\infty} \frac{n^2}{2}$)
- asymptotic notations (e.g., $\mathcal{O}(n \log(n) e^n)$ or $o(n^5)$)
- proofs by induction
- solving $u_{n+1} = 4u_n + 5$, $u_0 = 2$

Just to asess what you are already comfortable with

Raise your hand if you are comfortable with

- long addition/division/multiplication
- solving linear equations (e.g. $4x + 5 = 7$)
- what are the difference between integers, rational numbers and reals?
- what is a (mathematical) function?
- what is a sequence of numbers?
- limits (e.g. $\lim_{n\to+\infty} \frac{n^2}{2}$)
- asymptotic notations (e.g., $\mathcal{O}(n\log(n)e^n)$ or $o(n^5)$)
- proofs by induction
- solving $u_{n+1} = 4u_n + 5$, $u_0 = 2$
- solving $f(n+1) = 3f(\lceil \frac{n}{2} \rceil) + n^2$, $f(0) = 2$

Some extrinsic motivation: job interview questions

The maths has a practical point. Here are some typical job interview questions you should be able to answer after having taken the module:

- Find the largest element in an array
- What is a linked list?
- Calculate the length of a linked list
- Reverse a singly linked list
- Would you use a linked list or a dynamic array to store a list of actors in a video game?
- Compute the sum of all the values in the nodes of a linked list
- Print the values in a singly linked list in reverse order
- Write insert and remove operations for a singly linked list
- How would you sort a singly linked list?
- What are the advantages and disadvantages of linked lists vs. arrays?

More interview questions

- What is a binary search tree?
- Implement an enumerator for binary tree traversal
- Write a function to search a BST for a key and return the found node
- When would you use a binary tree vs. a hash table?
- Write a function which will insert a number into a BST
- Write a function which will flatten a binary tree into an array
- Write a function which will link the siblings in a tree
- Write a method to set all the data in a tree nodes to the empty string.
- How fast can you compute the distance between two nodes in a tree-like graph?
- Implement a queue using two stacks

Week-by-week provisional plan

- Week 1 & 2: what is an algorithm

  (high-level explanation, basic examples, pseudocode)
- Week 2 & 3: complexity
- Week 3 & 4: standard problem-solving techniques

  (recursion, divide-and conquer, dynamic programming)
- Week 5: introducing datastructures, arrays vs linked lists
- Week 6: stacks, queues
- Week 7 & 8: tree-like structures
- Week 9: graphs
- Week 10: hashing

Lectures

Thursdays noon-2PM

- 5 min breaks in the middle?
- Will attempt to record and publish all materials, but do come in

slides/virtual boards are not *everything*

Week-by-week updates on canvas

Tuesday at the latest after the lecture, lab material uploaded on Fridays

Week 9 and 10

I will be away in some castle in Germany to do research.
Will be replaced for lectures (probably by Oliver Kullman)

Labs/tutorials

Monday 3-5PM, CoFo 104

- There will be machines in the room...
- ...however you will not necessarily do much programming
- **please bring pen & paper** (or any note-taking device)
- If using your laptop, please have java set up properly

(I suppose you have done CSCM41)

Teaching material

I will at least publish on canvas

- Any lecture/tutorial material
- Coursework handouts
- Slides from my predecessor
  - Sometimes I will use them in lectures
  - Please do not disseminate those in particular

Bibliography:

- R. Sedgewick and K. Wayne, *Algorithms* (4th ed., 2011).
- R. Sedgewick, *Algorithms in Java* (3rd ed., 2004).
- T. Cormen et. al., *Introduction to Algorithms* (4th ed., 2011).
  Not on the reading list, but my go-to textbook

Context/other learning resources

The syllabus is rather standard

- Content: not beyond a bachelor CS program
- (summary in a revision sheet on canvas)
- Consequence: lots of resources in the wild

- You are of course free to learn the material however you want...
- Be careful with not-so-reliable online resources
  - Q&A website such as StackExchange
  - "Learning platforms" that are content-farmy/contain "AI" garbage

(geekforgeeks, anything that serves you ads)

Context/other learning resources

The syllabus is rather standard

- Content: not beyond a bachelor CS program
- (summary in a revision sheet on canvas)
- Consequence: lots of resources in the wild

- You are of course free to learn the material however you want...
- Be careful with not-so-reliable online resources
  - Q&A website such as StackExchange
  - "Learning platforms" that are content-farmy/contain "AI" garbage

(geekforgeeks, anything that serves you ads)

My anecdotal data

Geekforgeeks had a MCQ about datastructures asking to compute

$$\int_0^{\pi} x \sin(x) dx$$

(off-topic!! why are you asking??????)

and also all the answers were obviously wrong???

Cop slide

This is when I say cop stuff. TL;DR:

- Using "AI" (LLMs) is a bad idea
  - to learn things (it does not know things)
  - to communicate

    (people usually resent you when receiving canned content-free walls of text)
- You could get correctish answers for some stuff
  - because, again, the content is **standard**
  - (we are sometimes *very* lenient with what counts as correct)

Assessment

30% Courseworks

- CW 1: pen & paper due March 5th
- CW2: Java programming, due March 19th
- Release: two weeks before the deadline
- A few bonus marks for engagement in tutorials before the due date

70% Exam in June

See past exams on canvas - usually people find it takes a bit of preparation.

Also see the revision sheet I put up for the previous year!

Questions

Challenges for me

- Variety of backgrounds in the cohort
- I have a very theory-heavy background (I do like maths)
- Make you into masters of computers!

So do not hesitate to

- Ask questions
- Ask me to explain/revisit some points
- Interrupt!
- Give feedback & suggestions in the discussion board on canvas

Questions

Challenges for me

- Variety of backgrounds in the cohort
- I have a very theory-heavy background (I do like maths)
- Make you into masters of computers!

So do not hesitate to

- Ask questions
- Ask me to explain/revisit some points
- Interrupt!
- Give feedback & suggestions in the discussion board on canvas

Any questions?