COMP1511 WEEK 10

Starting at 9:10am

Congrats for making it to Week 10!

- Any reflections on the course / term?
- Favourite parts?
- Things that weren’t so good?
- What are you grateful for?

Questions from the course?

- **Today**
  - ADTs and Stacks
  - A bit on recursion
  - Anything you want me to go over
  - Labs this week/final exam

Week 10 Lab

- Exam Environment
  - Part 1 (Short Answer/Multiple Choice Questions)
    - Not worth marks today
    - 20 questions in the final exam
  - Part 2 (Programming Questions)
    - First 3 lab exercises for this week found here
    - 8 questions in the final exam
- Normal lab exercises
  - 1 standard and 1 challenge exercise

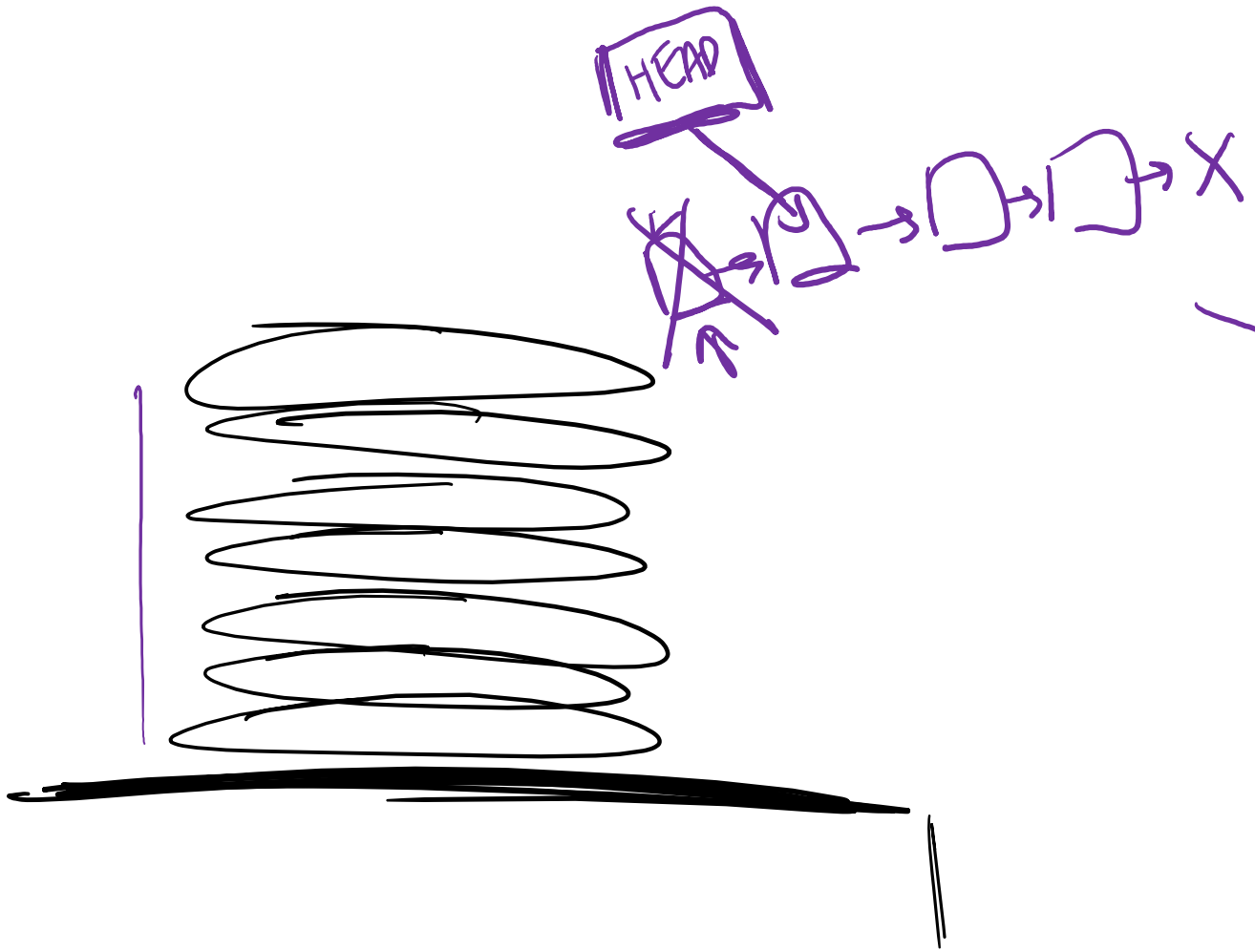
Final Exam

- Wed 18st August, 1pm – 7pm AEST
- Read-only open book, take-home exam (online)
- Expected to take 3-5 hours
- A staff member will always be available to contact
- Array + Linked List Hurdles

https://cgi.cse.unsw.edu.au/~cs1511/21T2/lec/exam/slides

ADTs, a concept – consider a QUEUE

Stack – last in, first out



What functions?

- create stack
- push to stack (add)
- pop from stack (remove)
- count items
- free stack

stack.c and stack.h

- Implementation – linked list

stack.c and stack.h

Header File

- *.h
- Typedef
- Function declarations
- Easy access to functions
- Don’t know how functions are implemented

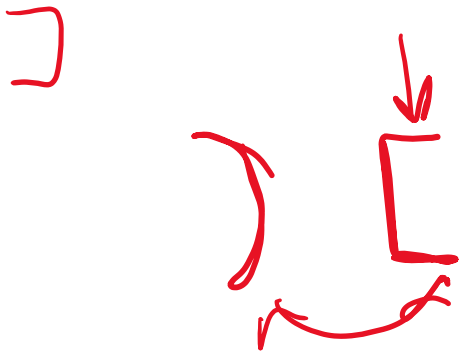
Implementation File

- *.c
- #include “header”
- Function implementation

Matching Brackets

[(){](){} invalid

[[]](){()} valid



Recursion

$5! = 5 \times 4 \times 3 \times 2 \times 1$

$5! = 5 \times 4!$

$4! = 4 \times 3!$

$3! = 3 \times 2!$

$1! = 1$

- A function that calls itself
  - Breaks itself down into smaller sub-problems
- How to solve any recursion question
  1. Write the base/terminating case
  2. Write the recursive case

$n! = n \times (n-1)!$

Fibonacci number

https://en.wikipedia.org/wiki/Fibonacci_number

From Wikipedia, the free encyclopedia

"Fibonacci Sequence" redirects here. For the chamber ensemble, see Fibonacci Sequence (ensemble).

In mathematics, the **Fibonacci numbers**, commonly denoted F_n, form a sequence, called the **Fibonacci sequence**, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is,[1]

$$F_0 = 0, \quad F_1 = 1,$$

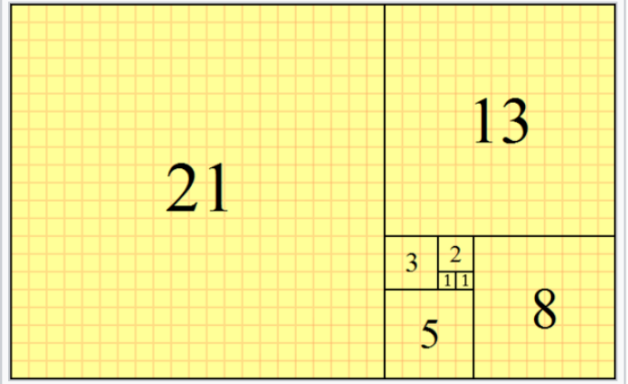
and

$$F_n = F_{n-1} + F_{n-2}$$

for $n > 1$.

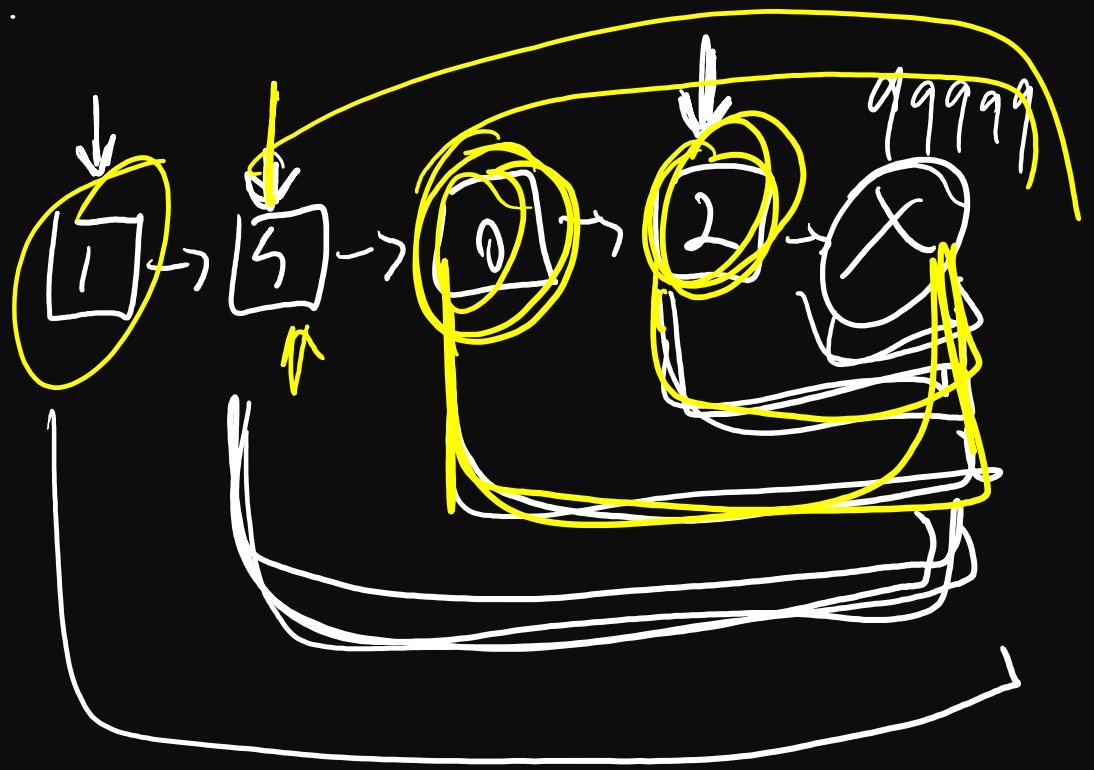
The sequence starts:[2]

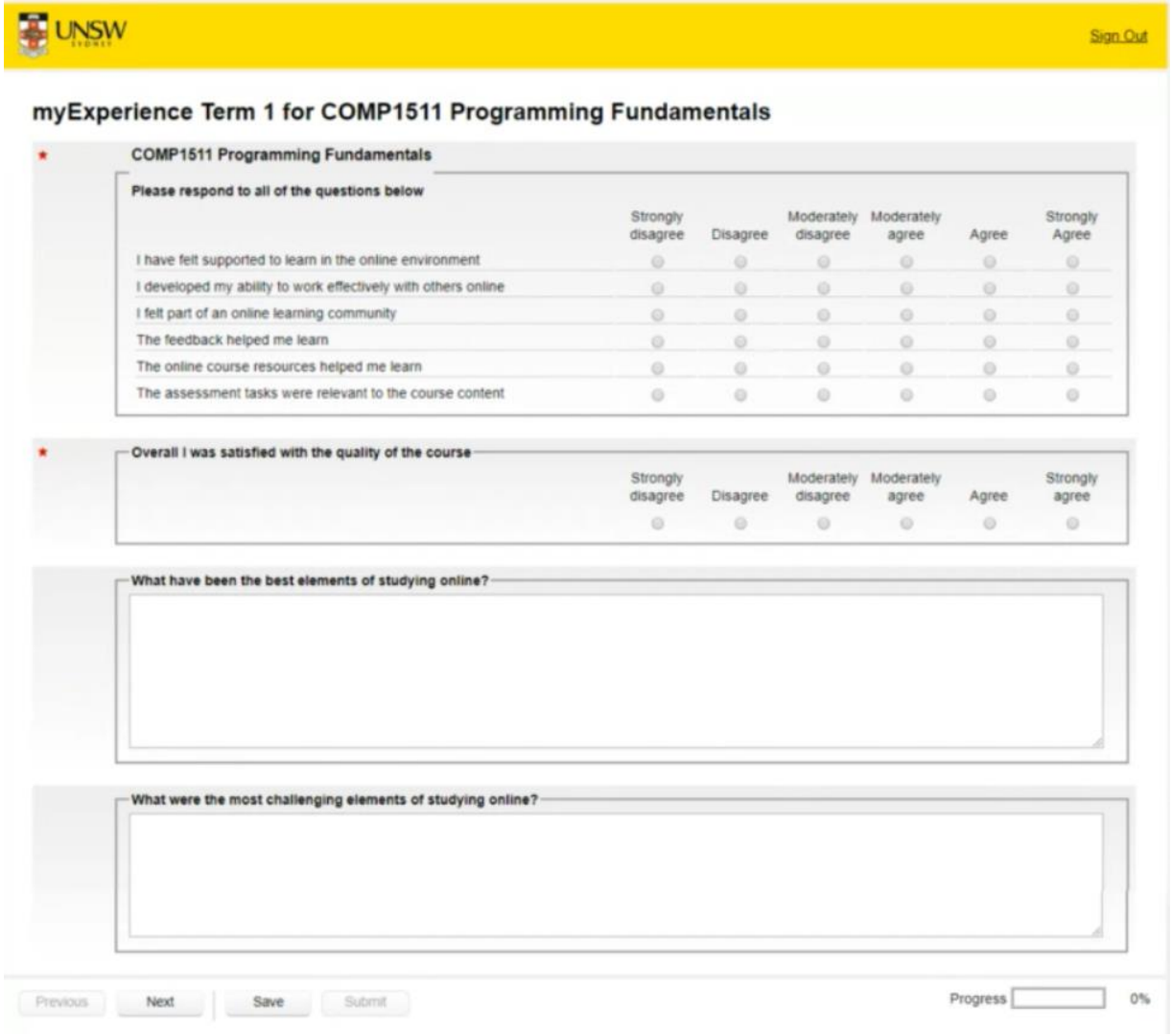
$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots$$



A tiling with squares whose side lengths are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13 and 21.

```c
int minimum(struct node* head) {

    if (head == NULL) {
        return INT_MAX;
    }

    int curr_min = head->data;
    int rest_min = minimum(head->next);

    if (curr_min < rest_min) {
        return curr_min;
    }

    return rest_min;
}
```





https://myexperience.unsw.edu.au/