Week 11 Studio Streams

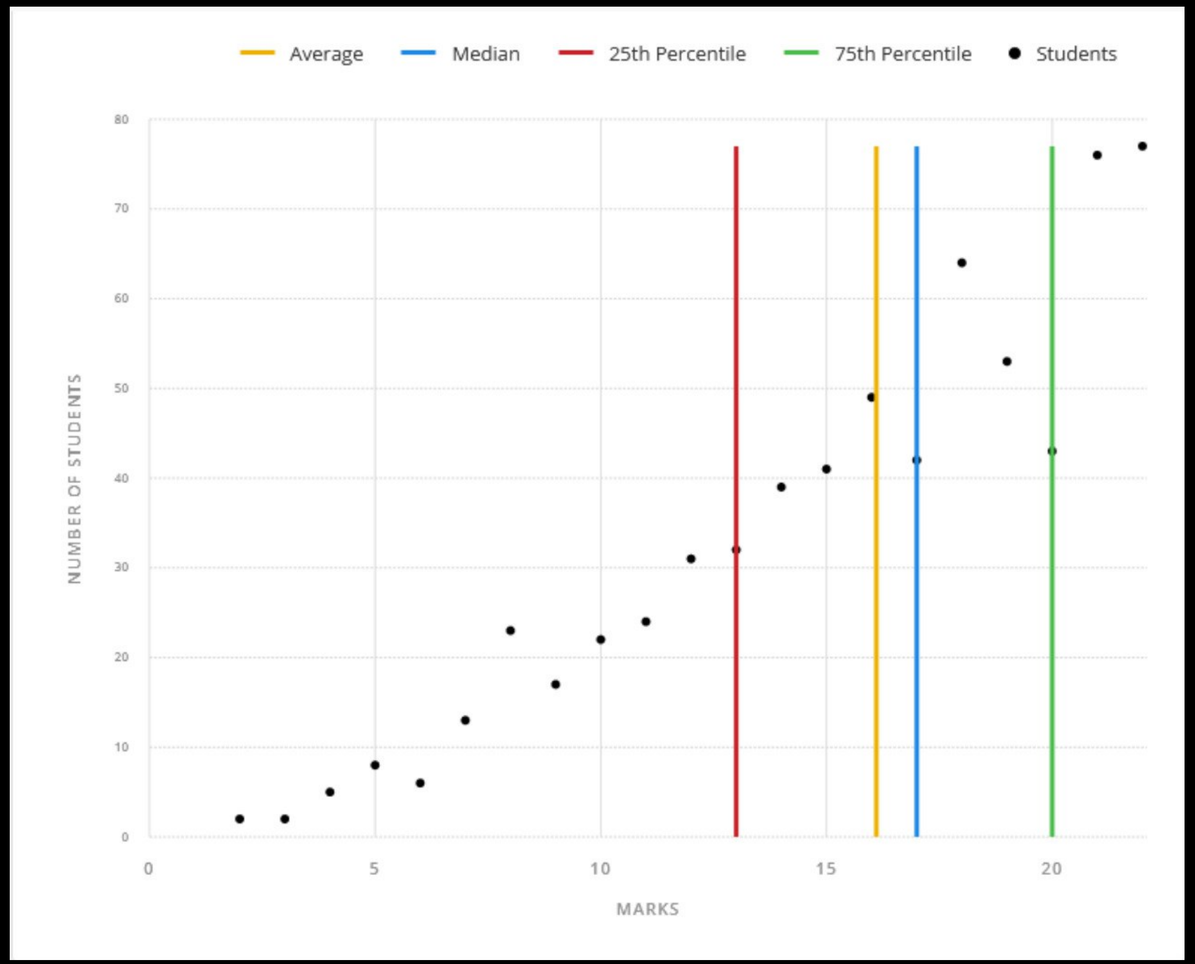
CS1101S AY21/22 Semester 1
Studio 05E

25 Oct 2021

Yan Xiaozhi (David)
@david_eom
yan_xiaozhi@u.nus.edu

Admin

- Contact tracing (QR code + class photo)
- Reading Assessment 2
  - Questions for me to go through!
- Practical assessment
  - Week 13 Wednesday? (3 Nov)
  - Redo your past missions and quests
  - Plan, then programme
  - Don’t give one-liner!



Recap

Streams

What Is It?

- A stream is either a
  - `null`, or
  - a pair whose tail is a nullity function that returns a stream
- Delayed list
  - Evaluation of tail is delayed until the function is being called
- Functional programming

Streams

What Is It?

- `pair(present, () => future)`
- Are these streams?
  - `null;`
  - `pair(1, () => pair(2, null));`
  - `pair(1, () => pair(2, () => null));`
  - `pair(1, () => pair(2, () => 3));`

Streams

Why?

- Ability to represent an infinite set of elements
  - How else to do an infinite list of integers or ones?
- Only compute when needed, less resource wasted
- Wishful thinking
- Common streams (try to implement them yourselves):
  - ones, integers, fibonacci, primes (a bit hard ah)

Streams

Useful Functions

- `stream_tail`
- `stream_length`
- `stream_ref`
- `stream_map`
- `build_stream`
- `eval_stream`
- `...`

- Check source documentation for what they do!
  - Useful during PE/finals
- Beware of their “laziness”
  - “Lazy? Yes/No/Sort-of”
  - Try `stream_length(ones);`
    - Prepare to force refresh your Source Acad man
  - Do NOT use non-lazy functions on infinite streams!

Streams

Stream of Streams

- ```
  function more(a, b) {
      return (a > b)
          ? more(1, 1 + b)
          : pair(a, () => more(a + 1, b));
  }
  ```
- `const more_and_more = more(1, 1);`
- What does `eval_stream(more_and_more, num)` return?
  ```

Streams

Techniques

- Use helper functions
  - Access the names declared
  - e.g.
```
const fib_helper =
            (a, b) => pair(a, () => fib_helper(b, a+b));
```
- Get rid of syntactic sugar
  - `function s() { return pair(1, s); }`
  - `function s() { return pair(1, () => s()); }`

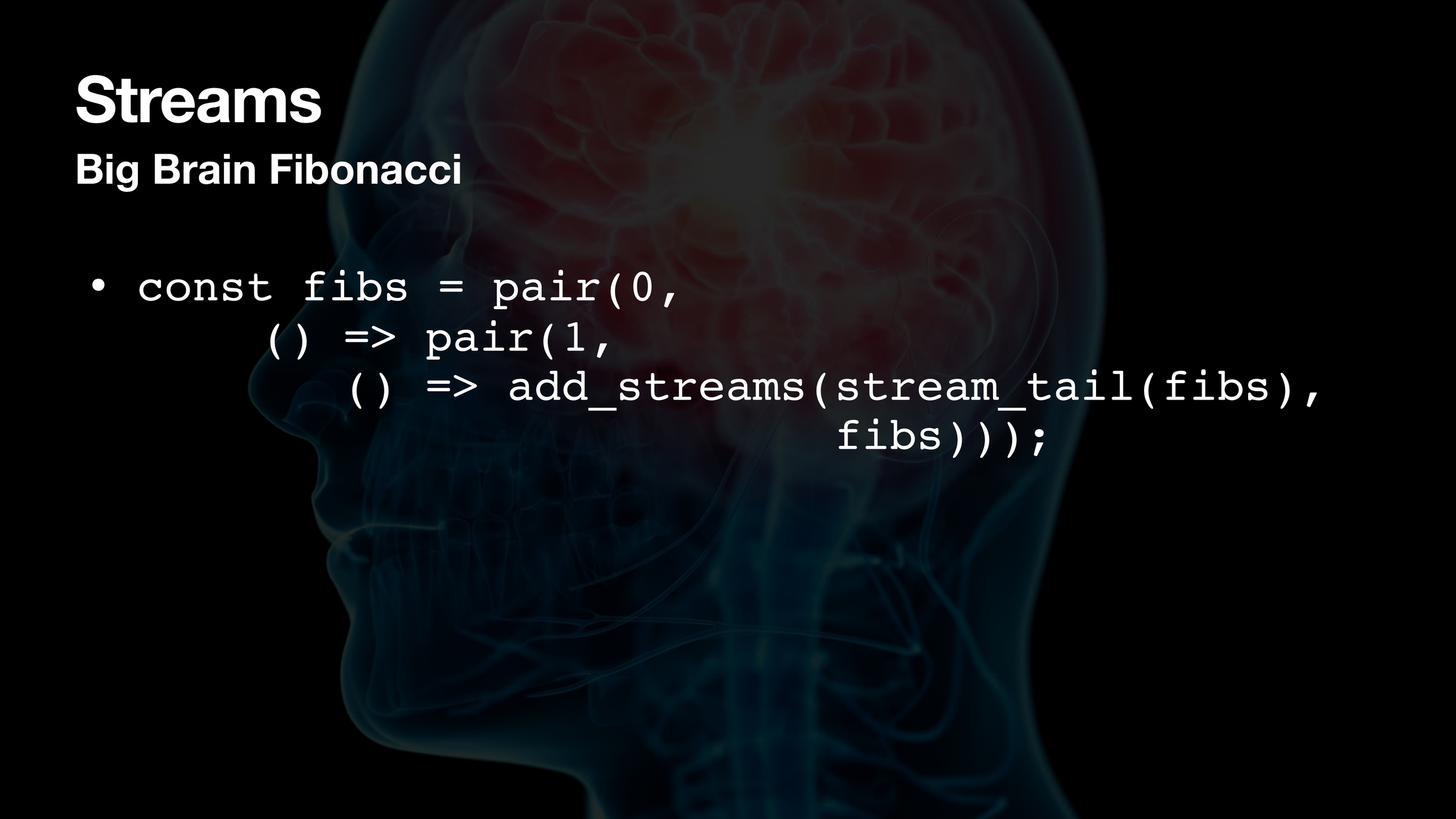
Streams

Techniques

- Play with multiple streams
-
```
function add_streams(s1, s2) {
    if (is_null(s1)) {
        return s2;
    } else if (is_null(s2)) {
        return s1;
    } else {
        return pair(head(s1) + head(s2),
                    () => add_streams(stream_tail(s1),
                                      stream_tail(s2)));
    }
}
```

Streams

Big Brain Fibonacci



- ```
  const fibs = pair(0,
        () => pair(1,
            () => add_streams(stream_tail(fibs),
                              fibs)));
  ```
  ```

Any Questions?