Ask me anything

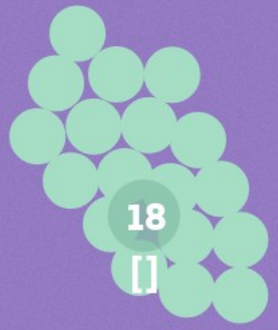
Mentimeter

0 questions
0 upvotes

Haskell Demo

What is the result of: head [] ?

Mentimeter



18

[]

1

nil

3

error

0

The horror, the horror ...

22

How can you be sure a recursive function will terminate?

Mentimeter

have a base case when they terminate

a base case that is achievable

it only terminates for finite lists thoughBut really: len xs is monotonically decreasing and there' s a base case

reduce problem with each step

proof by induction

5

How would you re-write fibs so that (a+b) only appears once?

Mentimeter

is this correct?fibs = 1 : 1 : fibsFollowing 1 1where fibsFollowing a b = ((a+b) = c) : fibsFollowing b c

fiboCont :: Integral a => a -> ((a, a) -> a) -> a fiboCont 0 cont = cont (0, 0) fiboCont 1 cont = cont (1, 0) fiboCont n cont = fiboCont (n-1) (\(x, y) -> cont (x+y, x))

fib 0 s = sfib 1 s = 1+sfib n s = n + fib (n-1) 0

3

```
fibs' = fibsFrom 1 1
  where fibsFrom a b =
    a : fibsFrom b (a+b)
```

How would you write a tail-recursive Fibonacci function?

fib s i n = if i == n then s else fib (s+i) (i+1) n

fiboCont 0 cont = cont (0, 0) fiboCont 1 cont = cont (1, 0) fiboCont n cont = fiboCont (n-1) (\(x, y) -> cont (x+y, x))

fib 0 s = sfib 1 s = x+1fib n s = n + fib (n-1) scall: fib n 0

3

```
fib'' n = trFib (n-1) 1 1
    where
        trFib 0 _ j = j
        trFib n k j = trFib (n-1) (k+j) k

-- idea: trFib i (fib (j+1)) (fib j) == fib (i+j)
```

Mentimeter

Last chance for questions