

We Need the Size of Each Thing

What new information will our sort need?

First, how big is a “thing?”

We’ll **pass** the answer


- as a **number of bytes**
- **in a parameter.**

thing
thing
thing
thing
thing
thing

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 9

9

Help Prof. Lumetta ... Again



Which horse is first?

But to sort “things,” we need to have an order!

The solution? A callback!

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 10

10

What Signature Do We Use for the Callback?

What is the signature of the callback function?

We need to pass it two “things,” **t1** and **t2**.*

Let’s use **void*** for those.

We get back an answer:

- **t1 < t2**, or
- **t1 ≥ t2**.

*Apologies to Dr. Theodor Geisel.

t1 →	thing
	thing
	thing
	thing
	thing
t2 →	thing
	thing

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 11

11

Function Signature for Our **isort** Function

```
int32_t isort
(void* base,
int32_t n_elts,
size_t size,
int32_t (*is_smaller)
(void* t1, void* t2));
```

points to an array

number of things in array

number of bytes per thing

pointer to a function comparing two things

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 12

12