University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

# ECE 220: Computer Systems & Programming

Dynamic Allocation Think-Pair-Share

## Moving Data Structures Requires Flattening

As you know,

- **pointers** are memory addresses
- and **don't mean anything**
  - **on other computers, nor**
  - **in a later execution** of the same program.

When a program wants

- **to save a data structure** to a file,
- or **to send a data structure to another computer**,
- it must **flatten the structure**.

## Flattening Means Packing into an Array of Bytes

To **flatten a data structure**,

- all **pointers must be removed**
- and the **data packed into a contiguous array of bytes**
- in a way that **allows the data structure to be rebuilt (unflattened)**.

Let's do an example of unflattening …

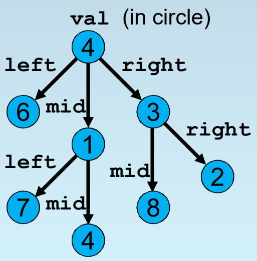
…as a think-pair-share.

But first, we'll do flattening together.

## Example: Flatten the Tree Shown Here

The node structure for the tree to the right:

```
struct node_t {
    node_t* left;
    node_t* mid;
    node_t* right;
    int32_t val;
};
```



Flattening can be done in any order. Let's use the order in the structure.