

Pack the Three Subtrees Recursively

Next, we write the three subtrees recursively.

On failure, we also fail.

```
if (-1 == (pos = pack_tree
    (ar, len, pos, root->left)) ||
    -1 == (pos = pack_tree
    (ar, len, pos, root->mid)) ||
    -1 == (pos = pack_tree
    (ar, len, pos, root->right))) {
    return -1;
}
```

Only called if
first call succeeds.

In which case, position
is that returned from
the first call.

Pack the Three Subtrees Recursively

Next, we write the three subtrees recursively.

On failure, we also fail.

```
if (-1 == (pos = pack_tree
    (ar, len, pos, root->left)) ||
    -1 == (pos = pack_tree
    (ar, len, pos, root->mid)) ||
    -1 == (pos = pack_tree
    (ar, len, pos, root->right))) {
    return -1;
}
```

Control flow and data between
second call and third call is exactly the same.

Finally, Write the Node's Value

```
if (len <= pos) {
    return -1;
}
```

Enough space
to write
value?

```
ar[pos] = root->val;
return (pos + 1);
```

Add value
to end of array.

Indicate that another space has been used.

Time for Another Think-Pair-Share

As before, let's do a group exercise in lecture.

The process:

1. I give you a problem.
2. You form groups of 3-4 people.
3. Talk about ways to solve the problem.
4. Once enough of the groups have finished, one group volunteers to share their answer.
5. We go over the group's answer together.