



# **Data Structure**

## **Lab Session #7: Non-Binary Trees**

**U Kang**  
**Seoul National University**



# Goals

- Implement **“Parent Pointer Tree”**
  - We recommend you to implement with following steps
    - `find()`
    - `differ()`
    - `groupSize()`
    - `union()`
- Solve **“Island Count Problem”**



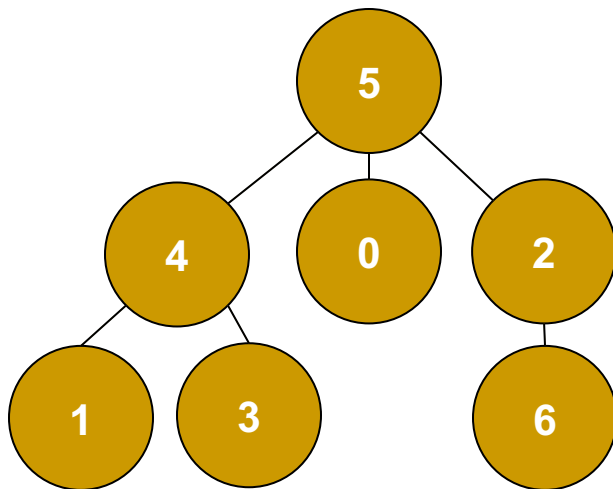
# Build a project

- Download the project for this lab from eTL.
- Extract the project, and import it using IntelliJ



# Parent Pointer Tree

- N-ary tree data structure in which each node has a pointer to its parent node, but no pointers to child nodes.



Index = node value

0	1	2	3	4	5	6
---	---	---	---	---	---	---

Array

5	4	5	4	5	null	2
---	---	---	---	---	------	---

cf) A value of the array is index of parent



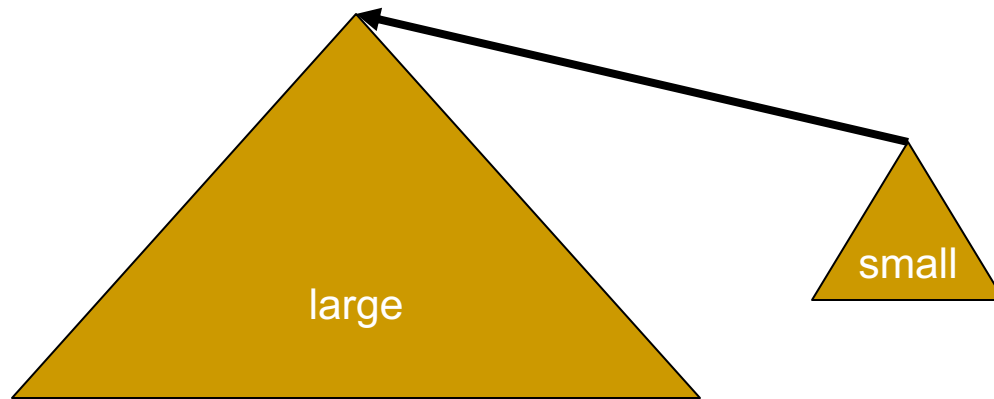
# Path compression

- Path compression is a way of flattening the structure of the tree whenever *find()* is used on it.
- When *find()* is called for an element  $x$ , root of the tree is returned.
- Make the found root as parent of  $x$  so that we don't have to traverse all intermediate nodes again.



# Weighted Union Rule

- Weighted union is joining the tree with fewer nodes to the tree with more nodes.





# I/O Specification

## ■ find

Input format	Output format
<b>find (node)</b>	
Description	
<ul style="list-style-type: none"><li>- Return a (root) node of tree which owns the (node).</li><li>- The method should perform path compression for the given (node).</li></ul>	
Example Input	Example Output
<b>find((node))</b>	<b>(root)</b>



# I/O Specification

## ■ differ

Input format	Output format
<code>differ (node1) (node2)</code>	
Description	
<ul style="list-style-type: none"><li>- Check whether the root nodes of the (node1) and (node2) are identical or not.</li><li>- (node1) and (node2) are the indices of the nodes.</li><li>- If two nodes are in the different group, return true</li><li>- If not, return false</li></ul>	
Example Input	Example Output
<code>differ((node1), (node2))</code>	<code>(boolean)</code>





# I/O Specification

## ■ groupSize

Input format	Output format
groupSize (node)	
Description	
<ul style="list-style-type: none"><li>- Print the size of the group that (node) belongs to.</li><li>- (node) is the index of the node as integer.</li><li>- (groupSize) is the size of the group that (node) belongs to.</li></ul>	
Example Input	Example Output
groupSize((node))	(groupSize)



# I/O Specification

## ■ union (weighted union rule)

Input format	Output format
union (node1) (node2)	
Description	
<ul style="list-style-type: none"><li>- Join the tree of (node2) and the tree of (node1), but follow “<b>Weighted Union Rule</b>”.</li><li>- (node1) and (node2) are the indices of the nodes as integer.</li><li>- Note that there is no output for this input.</li></ul>	
Example Input	Example Output
union((node1),(node2))	



# Sample Input & Output

## <Input>

```
union 2 3      union 6 11
union 1 3      union 6 12
union 4 5      differ 2 7
find 5         groupsize 2
union 1 5      groupsize 7
find 5         union 2 8
union 6 7      find 5
union 6 8      differ 2 7
union 6 9      union 5 13
union 6 10     groupsize 7
```

## <Output>

```
Union  2 and  3  Union  6 and 11
Union  1 and  3  Union  6 and 12
Union  4 and  5  DIFFER: YES
   5's ROOT: 4   GROUPSIZE: 5
Union  1 and  5  GROUPSIZE: 7
   5's ROOT: 2   Union  2 and  8
Union  6 and  7   5's ROOT: 6
Union  6 and  8  DIFFER: NO
Union  6 and  9  Union  5 and 13
Union  6 and 10  GROUPSIZE: 13
```



# Island Count Problem

- Given 2d grid (map) consists of zero (sea) and one (land), find the number of islands which are concatenated.

0	1	0	1
0	0	1	1
1	1	0	1
1	1	0	1

**Sample map  
consists of 3  
islands!**

**Sample map**



# Island Count Problem

- Input & output
  - Input: “sample\_map”
  - Output: “Sample map consists of 3 islands.”
  - Input: “test\_map”
  - Output: “Test map consists of ?? islands.”
    - You have to find ?? !
- Hint
  - Use union & find methods in parent pointer tree!



# Optional

- Could you print the size of the islands in sorted order? (bigger to smaller)

0	1	0	1
0	0	1	1
1	1	0	1
1	1	0	1

**Sample map can  
be segmented  
into 5 – 4 – 1  
size of islands!**

**Sample map**



# Questions?