What are possible strategies to employ when building a “smart union”?

---

Smart Union Strategy #1: ____________________________
Idea: Keep the height of the tree as small as possible!

Metadata at Root:

After union(4, 7):

```
6 6 6 8 10 7 7 7 4 5
```

---

Smart Union Strategy #2: ____________________________
Idea: Minimize the number of nodes that increase in height.
(Observe that the tree we union have all their nodes gain in height.)

Metadata at Root:

After union(4, 7):

```
6 6 6 8 10 7 7 7 4 5
```

---

Smart Union Implementation:

```
DisjointSets.cpp (partial)
void DisjointSets::unionBySize(int root1, int root2) {
    int newSize = arr_[root1] + arr_[root2];
    if (arr_[root1] < arr_[root2]) {
        arr_[root2] = root1;
        arr_[root1] = newSize;
    } else {
        arr_[root1] = root2;
        arr_[root2] = newSize;
    }
}
```
Path Compression:

UpTree Implementation with a smart union function and path compression:

```cpp
int DisjointSets::find(int i) {
    if ( arr_[i] < 0 ) { return i; } 
    else { return find( arr_[i] ); } 
}
```

```cpp
void DisjointSets::unionBySize(int root1, int root2) {
    int newSize = arr_[root1] + arr_[root2];
    // If arr_[root1] is less than (more negative), it is the larger set; we union the smaller set, root2, with root1.
    if ( arr_[root1] < arr_[root2] ) {
        arr_[root2] = root1;
        arr_[root1] = newSize;
    } 
    // Otherwise, do the opposite:
    else {
        arr_[root1] = root2;
        arr_[root1] = newSize;
    } 
}
```

A Review of Major Data Structures so Far

<table>
<thead>
<tr>
<th>Array-based</th>
<th>List/Pointer-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Sorted Array</td>
<td>- Singly Linked List</td>
</tr>
<tr>
<td>- Unsorted Array</td>
<td>- Doubly Linked List</td>
</tr>
<tr>
<td>- Stacks</td>
<td>- Skip Lists</td>
</tr>
<tr>
<td>- Queues</td>
<td>- Trees</td>
</tr>
<tr>
<td>- Hashing</td>
<td>- BTree</td>
</tr>
<tr>
<td>- Heaps</td>
<td>- Binary Tree</td>
</tr>
<tr>
<td>- Priority Queues</td>
<td>- Huffman Tree</td>
</tr>
<tr>
<td>- UpTrees</td>
<td>- kd-Tree</td>
</tr>
<tr>
<td>- Disjoint Sets</td>
<td>- AVL Tree</td>
</tr>
</tbody>
</table>

An Introduction to Graphs

CS 225 – Things To Be Doing:

1. Theory Exam 3 final day is **today**
2. lab_heaps due Sunday, April 8th
3. MP6 released; Extra Credit deadline on Monday, April 9th
4. Daily POTDs are ongoing!