Theorem: The running time of buildHeap on array of size $n$ is: 

\[
\text{_________}.
\]

Last Class:
We proved, by induction, that:

\[
S(h) = 2S(h-1) + h = 2^{h+1} - 2 - h
\]

Today, let us finish up talking about running times:

Heap Sort

Algorithm:
1.
2.
3.

Running time?

Why do we care about another sort?

Disjoint Sets
Let $R$ be an equivalence relation on $us$ where $(s, t) \in R$ if $s$ and $t$ have the same favorite among:

\[
\{ __, __, __, __, __, __ \}
\]

Examples:

Building Disjoint Sets:
- Maintain a collection $S = \{ s_0, s_1, \ldots, s_k \}$
- Each set has a representative member.
- ADT:
  - `void makeSet(const T & t);`
  - `void union(const T & k1, const T & k2);`
  - `T & find(const T & k);`
Implementation #1:

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td></td>
<td>2</td>
<td>7</td>
<td></td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Operation: find(k)

Operation: union(k1, k2)

Example:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Implementation #2:

- We will continue to use an array where the index is the key
- The value of the array is:
  - -1, if we have found the representative element
  - The index of the parent, if we haven’t found the rep. element

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Example:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Implementation

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

What is the running time?

```cpp
void DisjointSets::union(int r1, int r2) {
}
```

CS 225 – Things To Be Doing:

1. Register for CS 225’s Final Exam!
2. Exam #9 (theory exam) is ongoing
3. MP6 due Friday, Nov. 17
4. lab_dictionary due Sunday, Nov. 12
5. Daily POTDs