#31: Disjoint Sets

November 8, 2017 · *Wade Fagen-Ulmschneider*

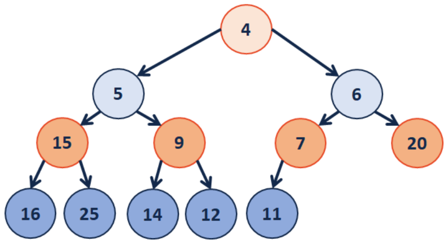
Theorem: The running time of buildHeap on array of size n is: __________.

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?

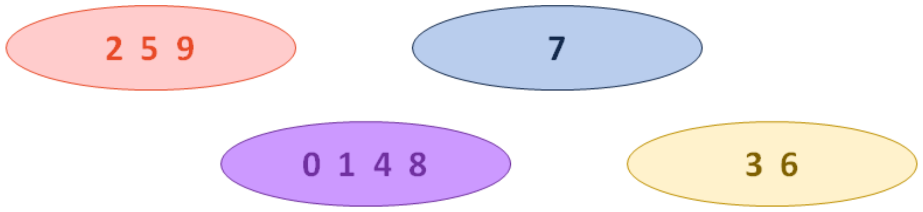
Reflections on Heaps

Disjoint Sets

Let **R** be an equivalence relation on *us* where **(s, t)** ∈ **R** if **s** and **t** have the same favorite among:

{ ____, ____, _____, ____, _____, ____ }

Examples:

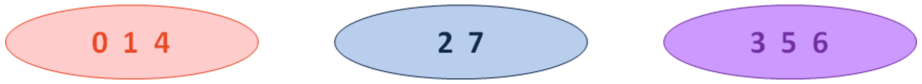


Building Disjoint Sets:

- 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:



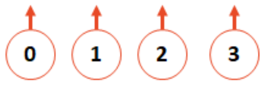
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

Operation: find(k)

Operation: union(k1, k2)

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



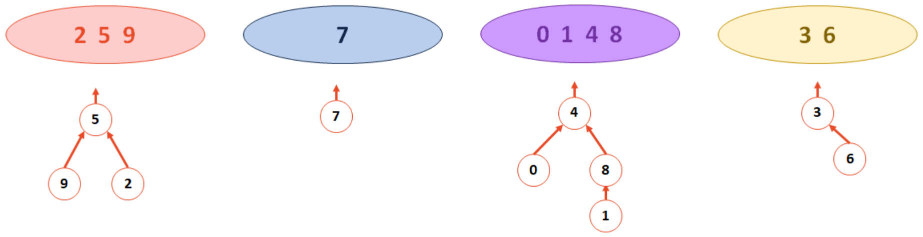
[0]	[1]	[2]	[3]

[0]	[1]	[2]	[3]

[0]	[1]	[2]	[3]

[0]	[1]	[2]	[3]

Example:



4	8	5	6	-1	-1	-1	-1	4	5
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

...what is the error in this table?

Implementation

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

What is the running time?

```
DisjointSets.cpp (partial)
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 onging
3. MP6 due Friday, Nov. 17
4. lab_dictionary due Sunday, Nov. 12
5. Daily POTDs