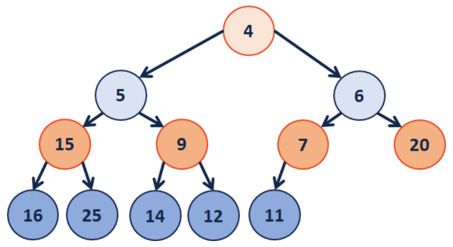
#29: Heap Operations

October 29, 2021 · *G Carl Evans*

A Heap Data Structure

(specifically a minHeap in this example, as the minimum element is at the root)

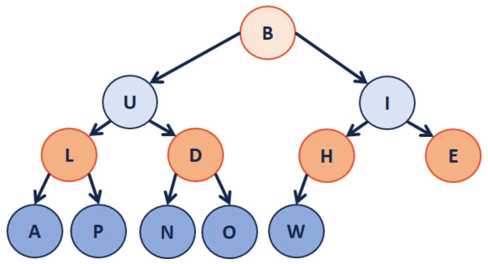


Heap Operation:

removeMin / heapifyDown

insert /heapifyUp

Q: How do we construct a heap given data?



-	B	U	I	L	D	H	E	A	P	N	O	W		

Ideas

1)

2)

3)

Running Time?

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

Strategy:

Define S(h):

Let **S(h)** denote the sum of the heights of all nodes in a complete tree of height **h**.

S(0) =

S(1) =

S(h) =

Proof of S(h) by Induction:

Finally, finding the running time:

Disjoint Sets

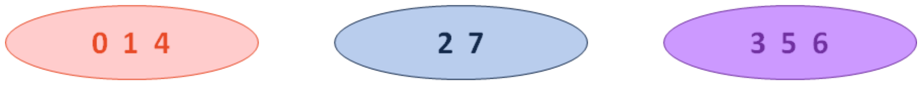
Let **R** be an equivalence relation. We represent R as disjoint sets

- Each element exists in exactly one set.
- Every set is an equitant representation.
  - Mathematically: $4 \in [0]_R \rightarrow 8 \in [0]_R$
  - Programmatically: find(4) == find(8)

Building Disjoint Sets:

- Maintain a collection $S = \{s_0, s_1, \dots s_k\}$
- Each set has a representative member

void makeSet(const T & t);
void union(const T & k1, const T & k2);
T & find(const T & k);



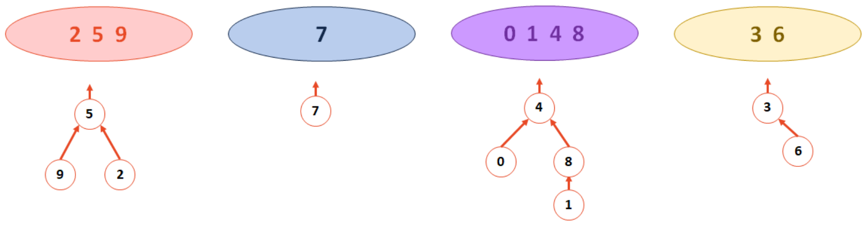
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

Operation: find(k)

Operation: union(k1, k2)

Implementation #2:

- 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



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

Implementation – DisjointSets::find

DisjointSets.cpp (partial)

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

What is the running time of **`find`**?

What is the ideal UpTree?

Implementation – DisjointSets::union

DisjointSets.cpp (partial)

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


}
```

How do we want to union the two UpTrees?

CS 225 – Things To Be Doing:

1. mp_mosaics due date: Monday, November 1
2. lab_hash is due Sunday, October 31
3. Daily POTDs are ongoing!