Heapify

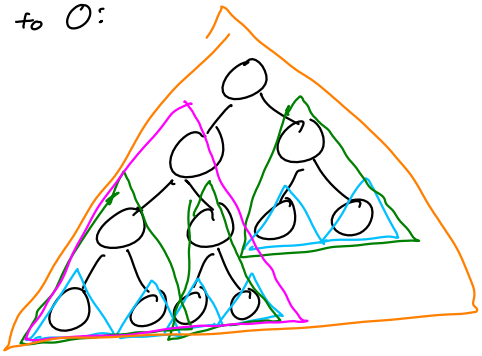
A CBT is a BT w/ all levels except last full, last level packed left

For any one size, there is one shape of CBT

Store all data in an array & compute indices

Algorithm: CBT → Heap

```
Function heapify(contents):
  For i in contents.length()-1 down to 0:
    bubbleDown(i)
  EndFor
EndFunction
```



Heapify is worst-case $O(n)$

- Half of the nodes are leaves
- 1/4th of the nodes are parents of leaves

# of steps:

$$1 \cdot \frac{n}{2} + 2 \cdot \frac{n}{4} + 3 \cdot \frac{n}{8} + \cdots + \log_2 n \cdot 1 \approx \sum_{i=1}^{\log_2 n} i \cdot \frac{n}{2^i}$$

$$\leq \sum_{i=1}^{\infty} i \cdot \frac{n}{2^i}$$

$$= n \cdot \boxed{\sum_{i=1}^{\infty} \frac{i}{2^i}}$$

$$\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \cdots$$

$$= \boxed{\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots} = 1$$

$$+ \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots \quad + \frac{1}{2}$$

$$+ \frac{1}{8} + \frac{1}{16} + \cdots \quad + \frac{1}{4}$$

$$+ \frac{1}{16} + \cdots \quad + \frac{1}{8}$$

$$+ \cdots$$

$$= 2$$

Most bubble-downs are small.

Hash Tables

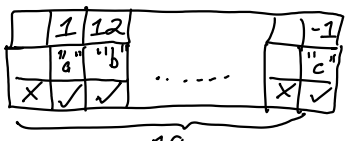
What is a dictionary? ADT

	AVL tree	Hash Table
V get(K)	O(log n)	average O(1)*
void insert(K, V)	O(log n)	average O(1)*
void update(K, V)	O(log n)	average O(1)*
V remove(K)	O(log n)	average O(1)*

Use an array; assume:

1. No collisions
2. Assume keys are integers
3. Assume no negative keys
4. Assume all keys < 10

(if array gets full, make a bigger one!)



insert(1, "a")
get(2)
insert(12, "b")
insert(-1, "c")

```
int hash(string s) {
  int acc = 0;
  for (int i=0; i<s.length(); i++) {
    acc *= 31;
    acc += s[i];
  }
  return acc;
}
```

okay "ba" "ab"

```
int hash(string s) {
  return s.length();
}
```

bad hash

```
int hash(string s) {
  return 0;
}
```

WORST HASH

Collision Resolution:

linear probing

key		1	5	
value		"a"	"b"	
in use?	X	✓	✓	X

insert(1, "a")
insert(5, "b")

forward chaining