CSCI 241

Scott Wehrwein

Heaps:
`peek` and `poll`

Goals

Be prepared to implement the `peek`, and `poll` heap operations and execute them on paper.

Heap Operations

```
interface PriorityQueue<V v, P p> {
 // insert value v with priority p
 void add(V v, P p);

 // return value with min priority
 V peek();

 // remove/return value with min priority
 V poll();

 // more methods…
}
```

`V peek();`

Algorithm:

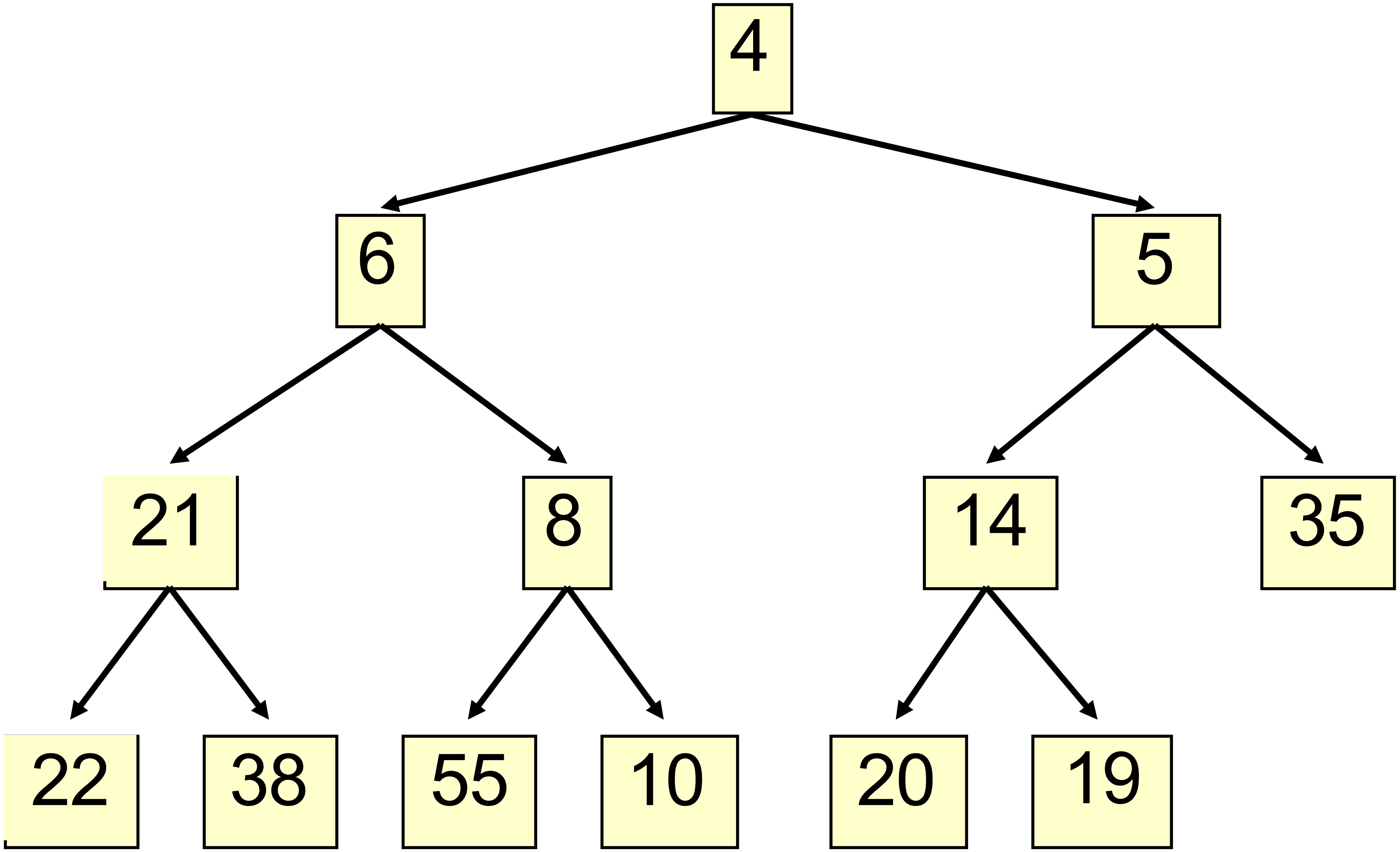
- Return the smallest thing.

`V poll();`

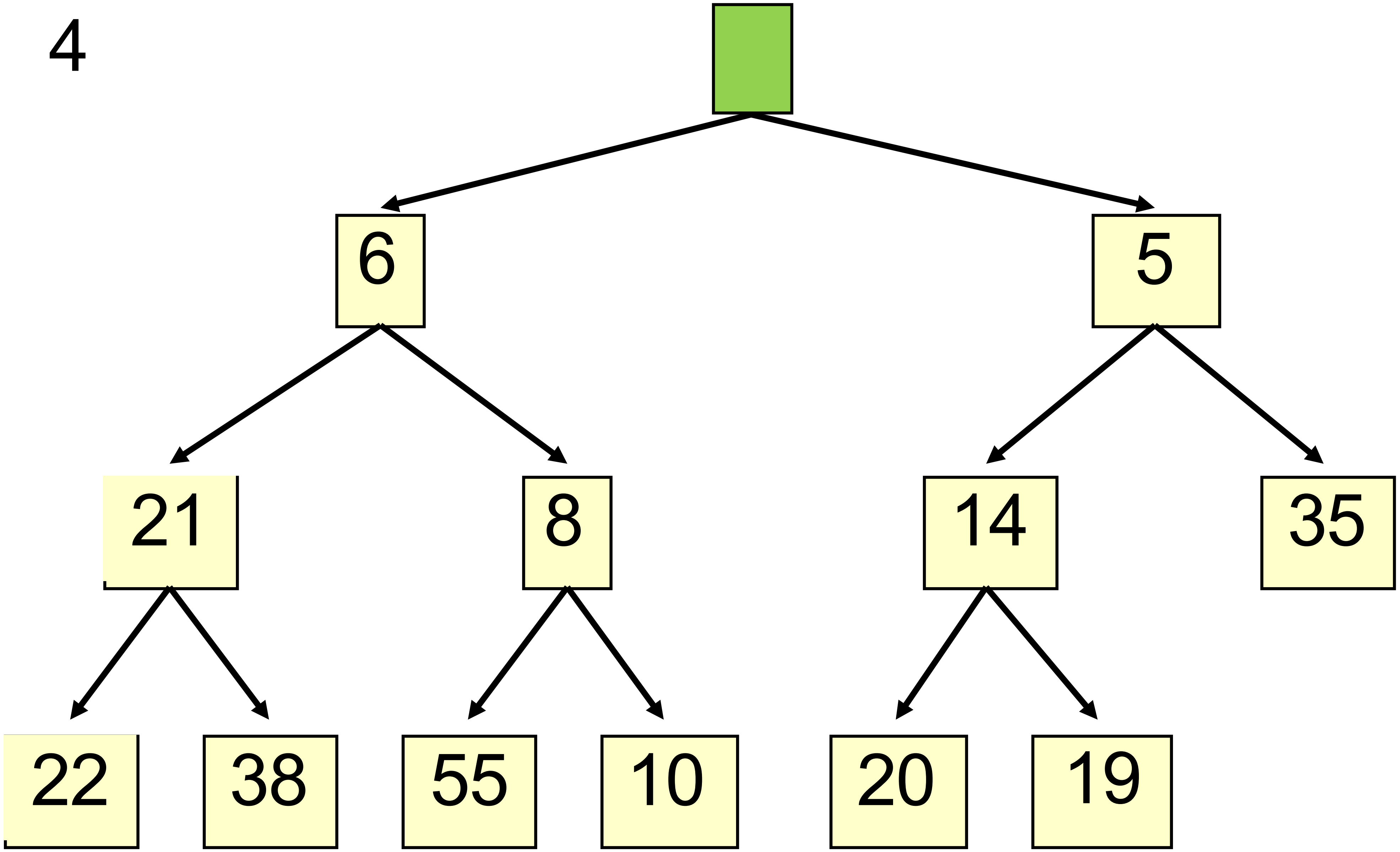
Algorithm:

- Remove and save the smallest thing
- Fill the resulting hole with the wrong thing
- Bubble the wrong thing down to the right place

```
V poll();
```

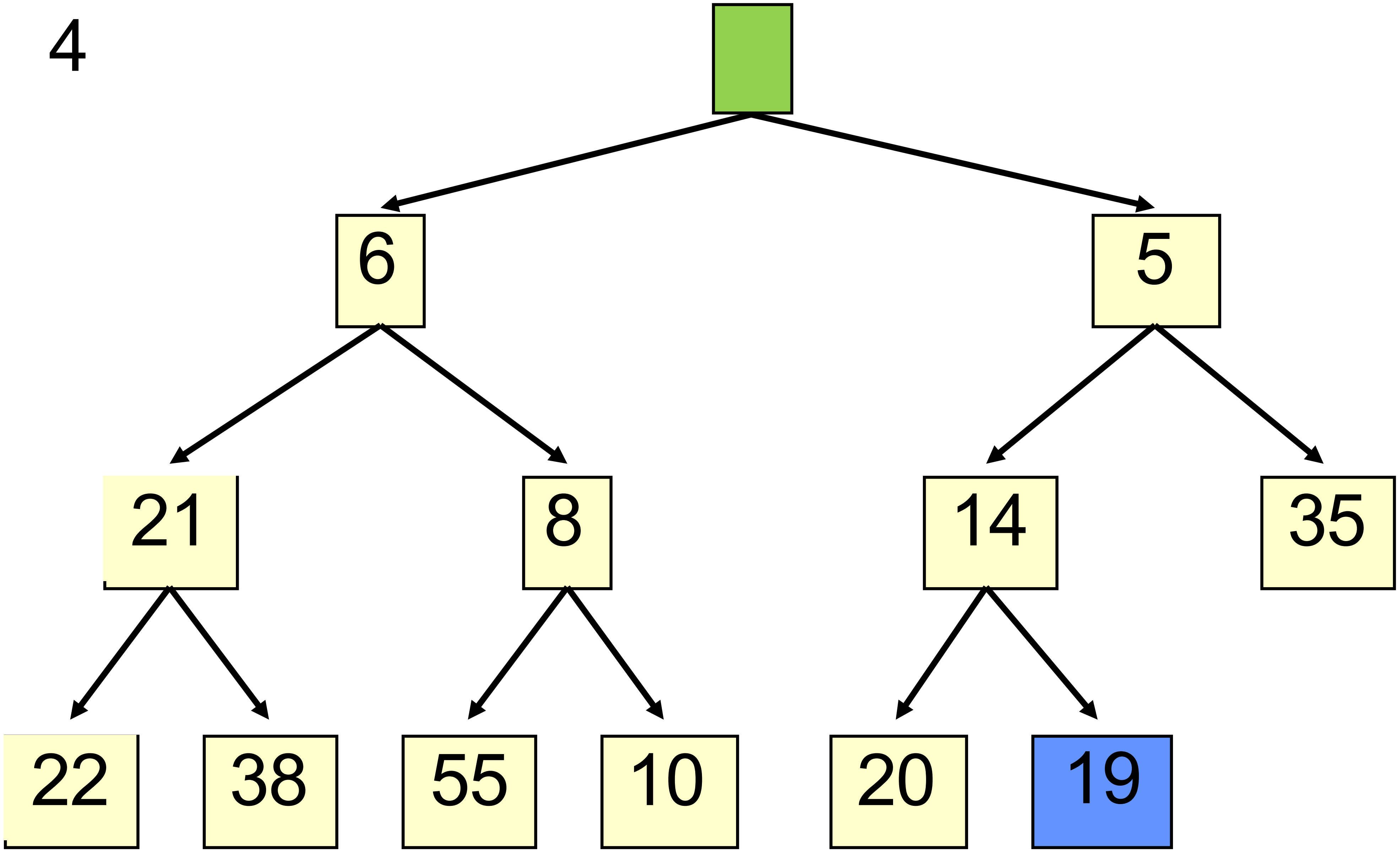


V poll();



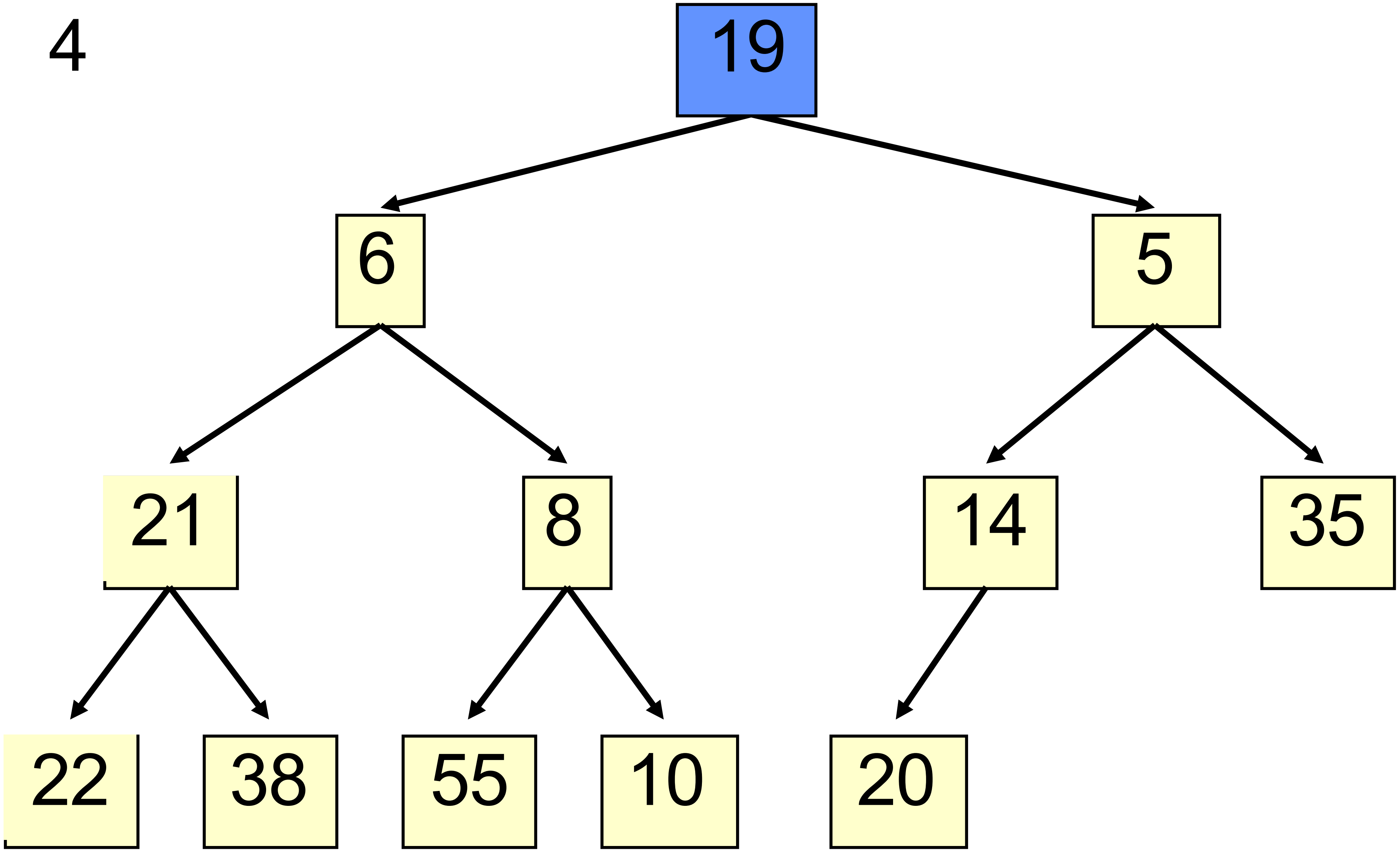
Remove and save the smallest (root) element

```
V poll();
```



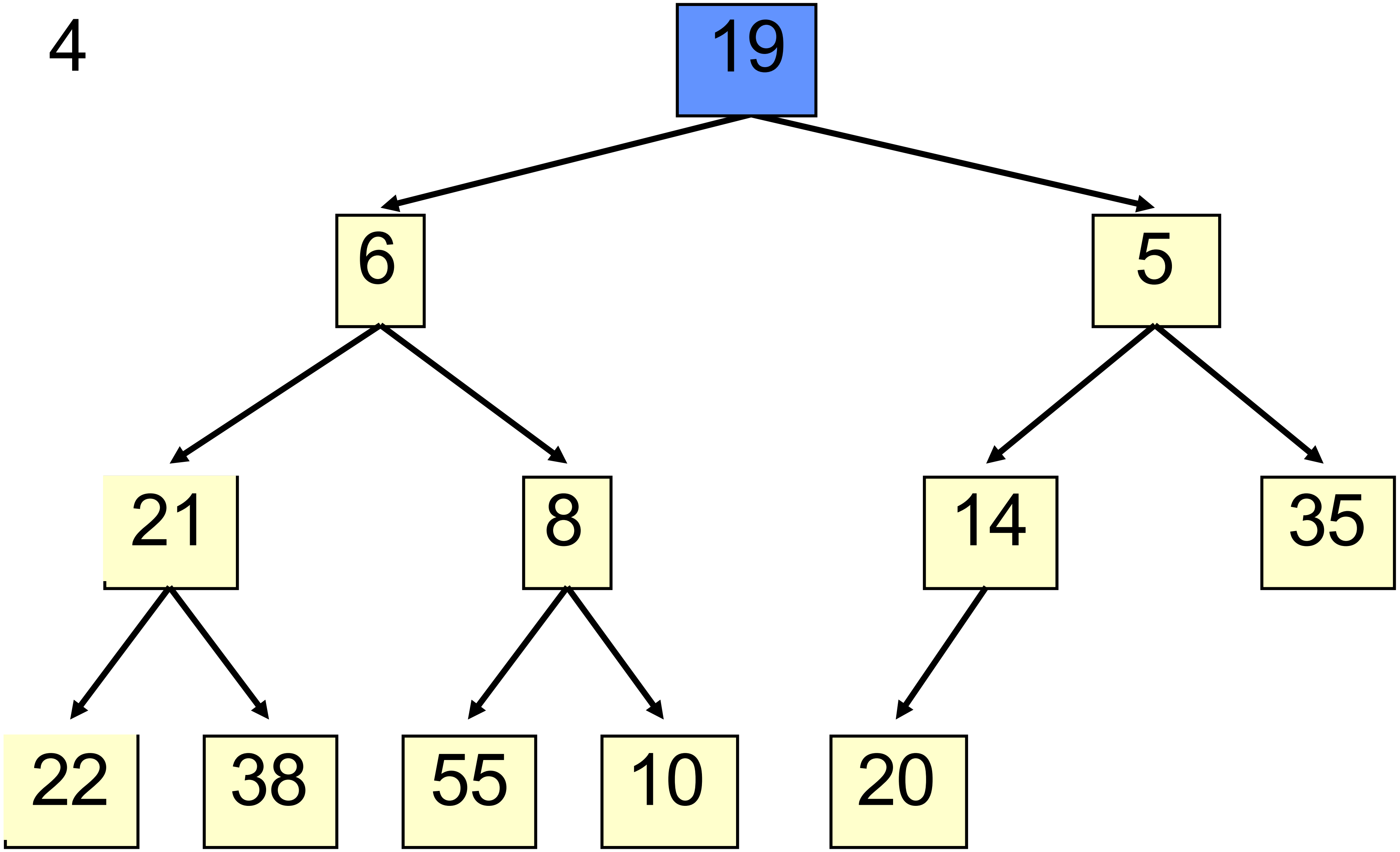
Move the last element to replace the root

`V poll();`



Bubble the root value down

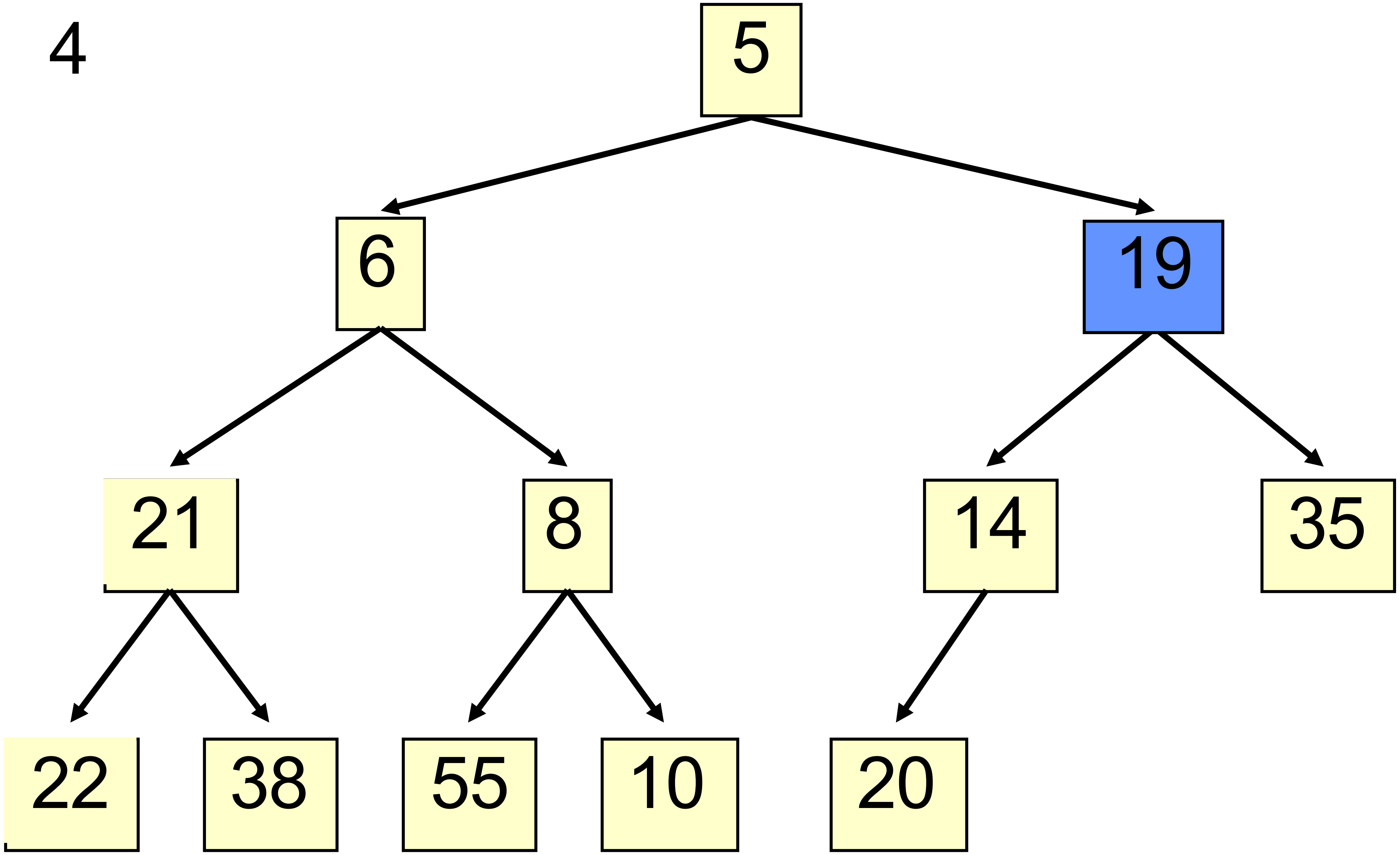
```
V poll();
```



Bubble the root value down, swapping with the **smaller** child

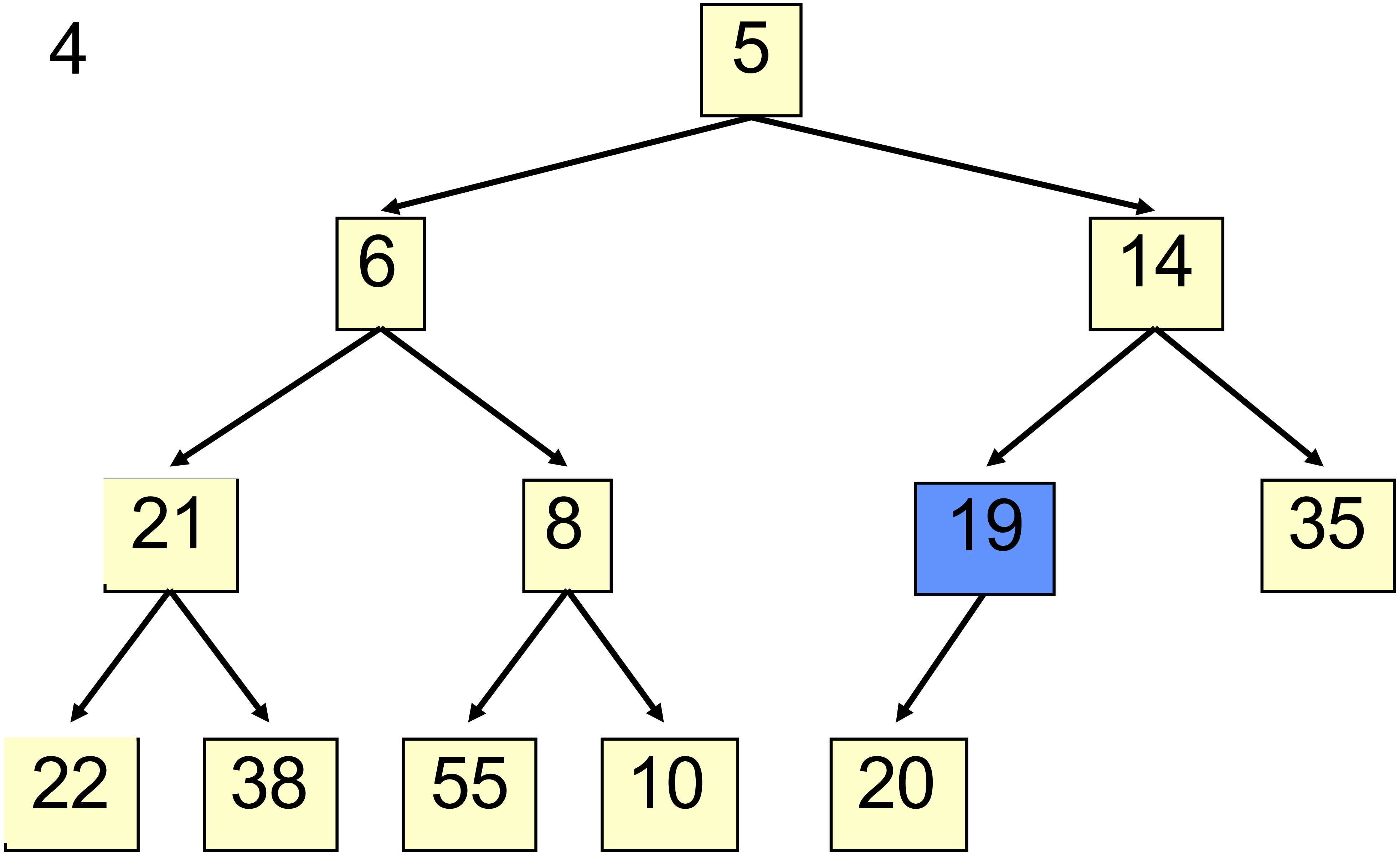
```
V poll();
```

4



Bubble the root value down, swapping with the **smaller** child

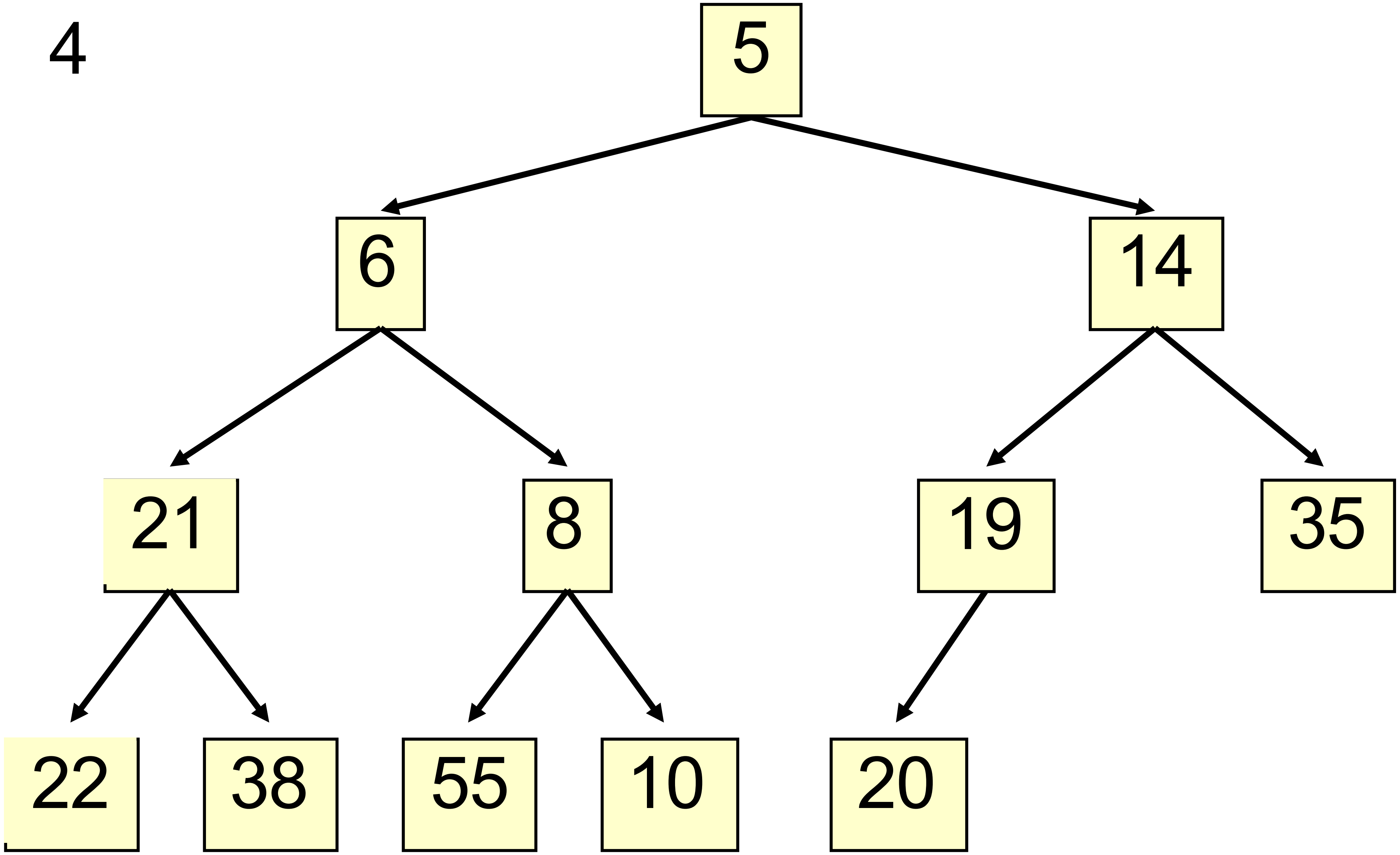
`V poll();`



Bubble the root value down, swapping with the **smaller** child

V poll();

4



Return the smallest element.

`V poll();`

Algorithm:

- Remove and save the root (first) element
- Move the last element to the first spot.
- While its priority is greater than either of its children's’:
    - Swap it with the child with smaller priority.