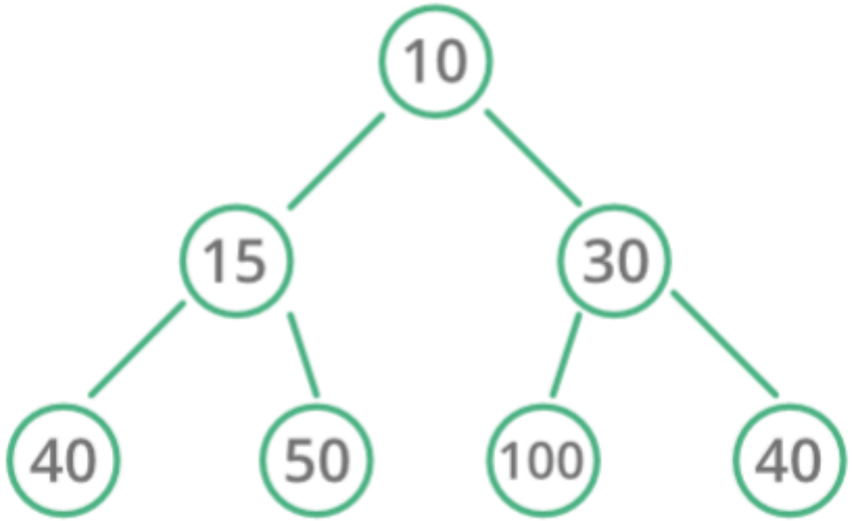
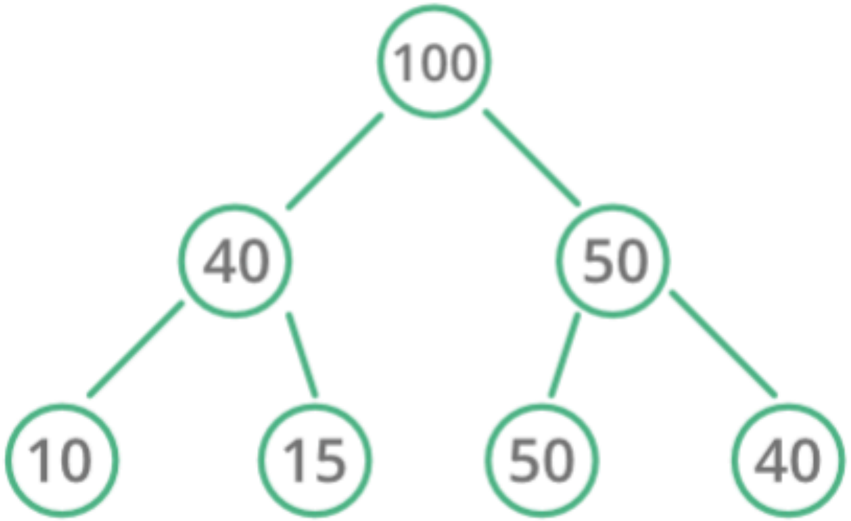
COSC2436: Heaps

Heaps

- **Max Heap**: The root must be the **maximum** among all of its children.
- **Min Heap**: The root must be the **minimum** among all of its children.



Min Heap



Max Heap

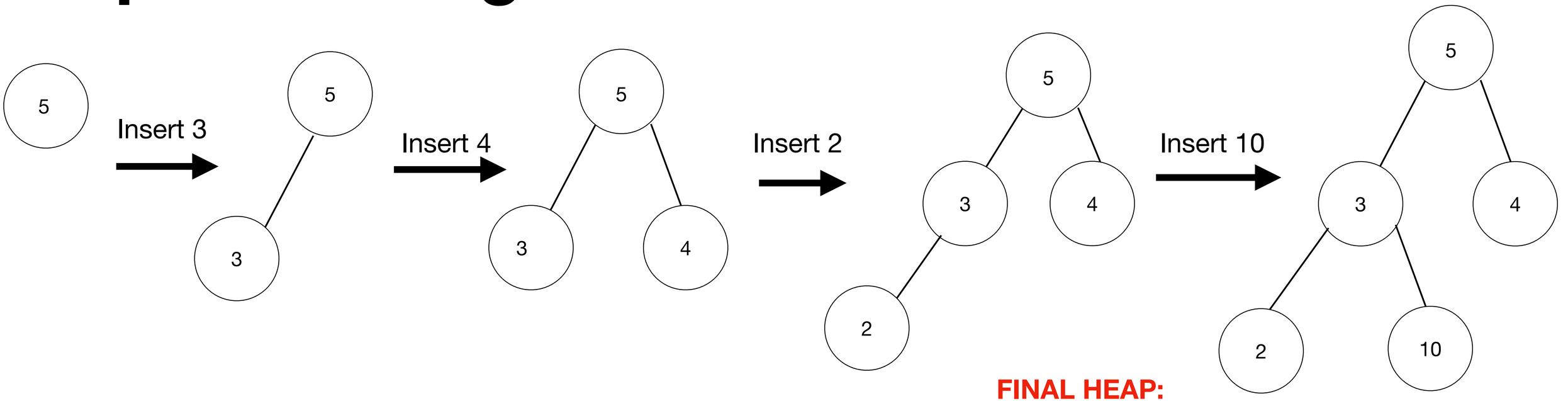
*The rest of this document will only use Max Hep

Heaps: Inserting

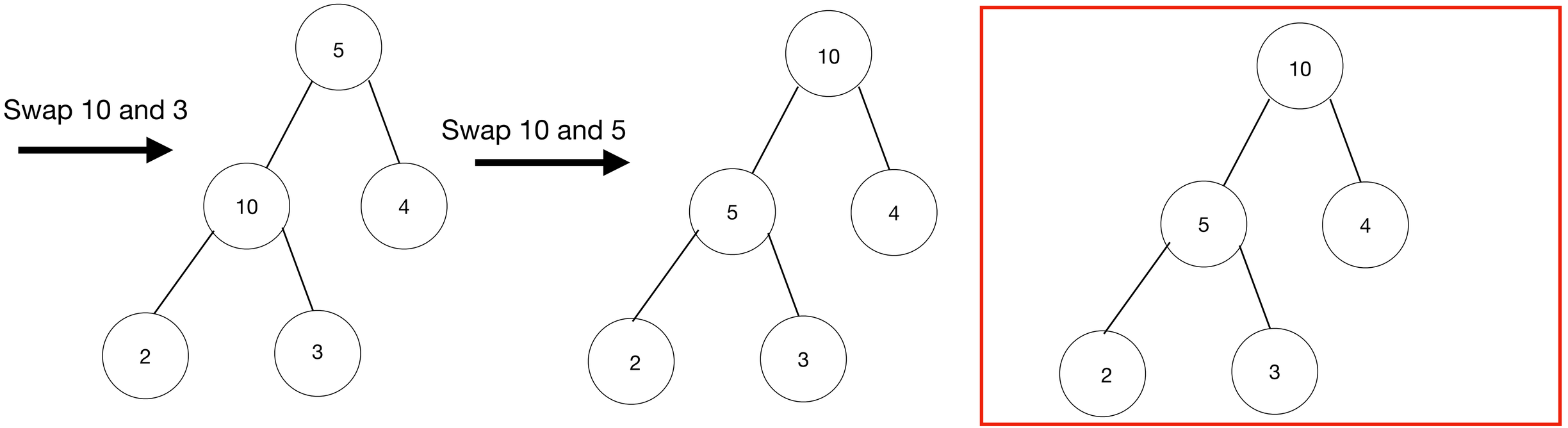
1. Insert key ***x*** at the bottom of the heap (never insert at the root of a heap)

2. Heapify

Heaps: Inserting



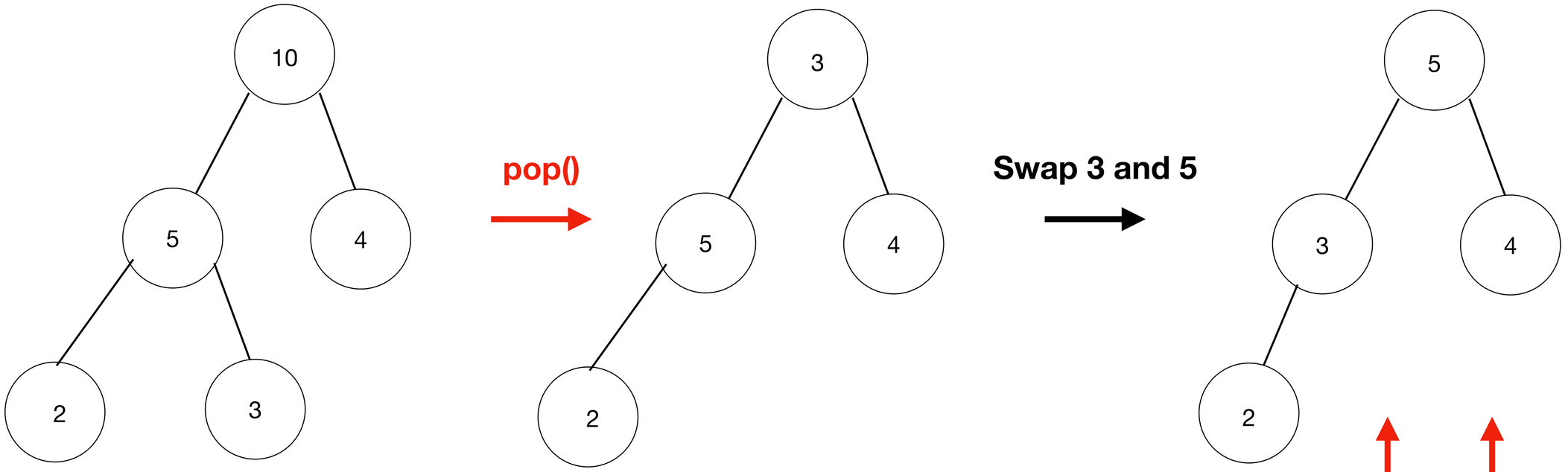
FINAL HEAP:



Heaps: Deletion

1. Remove key from root
2. Replace with the last key of the heap
3. Heapify

Heaps: Deletion



10 gets removed and replaced with 3

Heaps: Heapify

```
12 ▼ void heapify(int arr[], int n, int i){
13     int largest = i;
14     int l = 2 * i + 1;
15     int r = 2 * i + 2;
16
17 ▼   if(l < n && arr[l] > arr[largest]){
18       largest = l;
19     }
20
21 ▼   if(r < n && arr[r] > arr[largest]){
22       largest = r;
23     }
24
25 ▼   if(largest != i){
26       int temp = arr[largest];
27       arr[largest] = arr[i];
28       arr[i] = temp;
29
30       heapify(arr, n, largest);
31     }
32 }
```

Heaps: HeapSort

```
34 ▼ void heapSort(int arr[], int n){
35 ▼   for(int i = n / 2 - 1; i >= 0; i--){
36       heapify(arr, n, i);
37     }
38 ▼   for(int i = n - 1; i > 0; i--){
39       int temp = arr[i];
40       arr[i] = arr[0];
41       arr[0] = temp;
42
43       heapify(arr, i, 0);
44     }
45   }
```