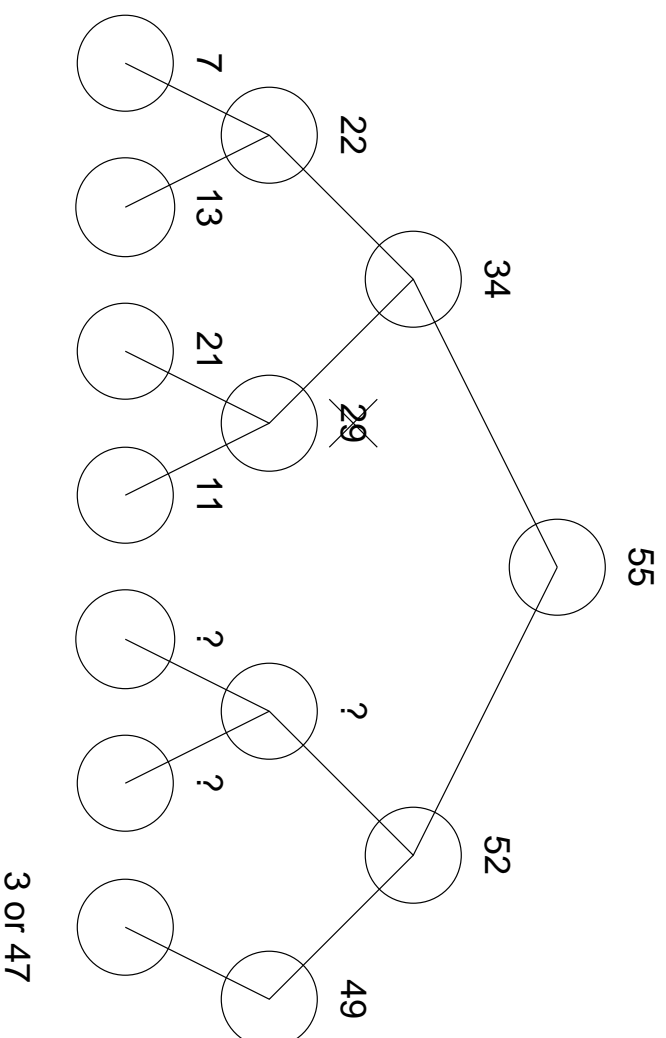# Overview

- Removal of a non-root node from a heap

- Quicksort

- Binary Trees
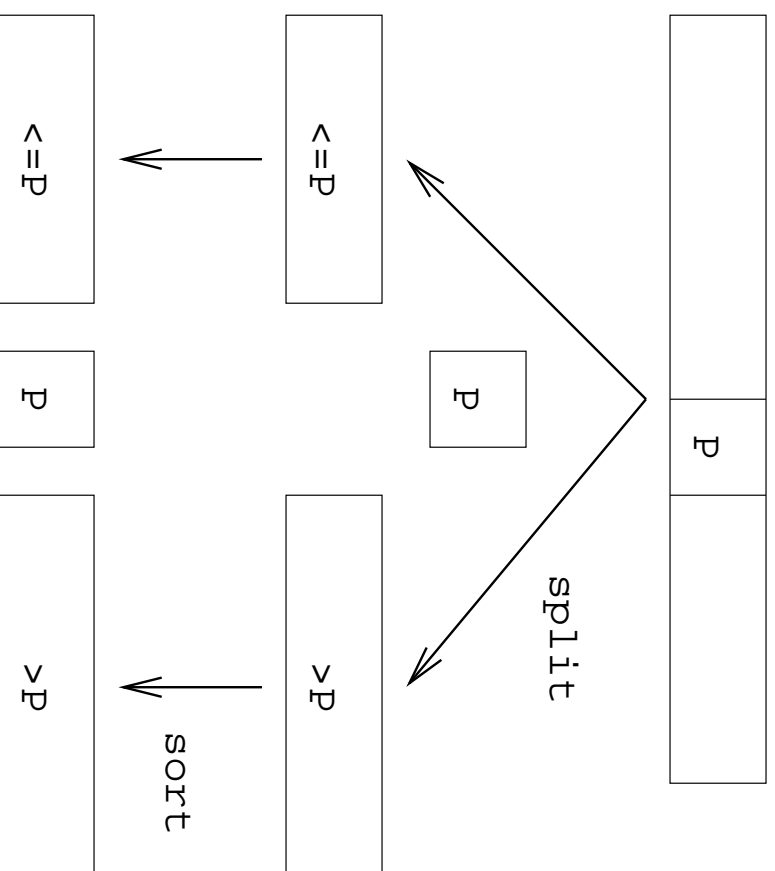
Removal Of A Non-root Node From A Heap

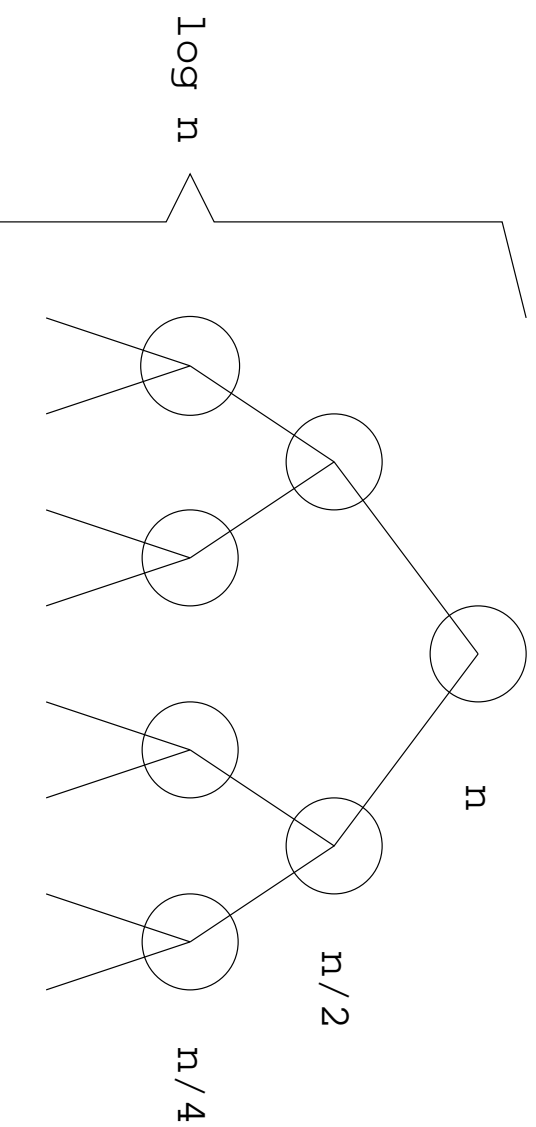- Consider removing the node with the key 29 from the following heap.

3 or 47

Quick Sort

- Quick Sort is a *hard-split, easy-join* method.

- The following diagram illustrate one step.

```
┌─────────┐
│    P    │
└─────────┘
     │
     │ split
    ╱ ╲
   ╱   ╲
┌──────┐   ┌──────┐
│ <=P  │   │  >P  │
└──────┘   └──────┘
   │          │
   │  sort    │
   ▼          ▼
┌──────┐ ┌─┐ ┌──────┐
│ <=P  │ │P│ │  >P  │
└──────┘ └─┘ └──────┘
```

Quick Sort (cont.)
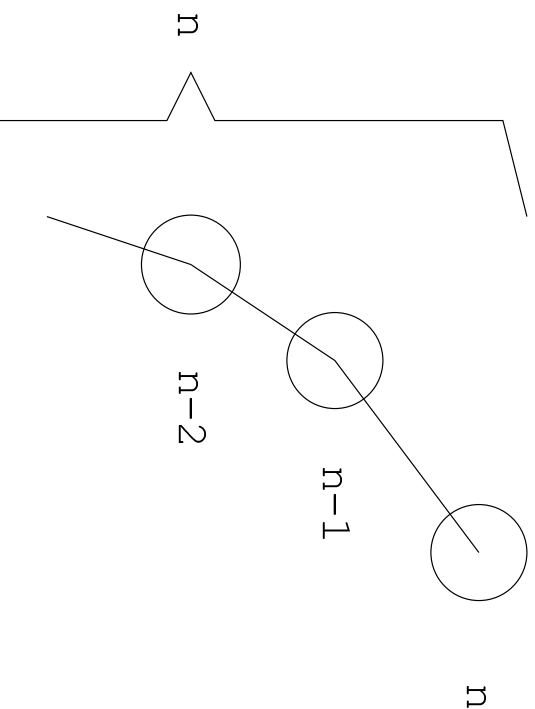
- If the pivot chosen by `split()` divides the array into two (almost) equal-sized parts, each element is `split()` $log\ n$ times.

log n

n

n/2

n/4

- Thus, in this case, Quick Sort takes $O(n\ log\ n)$ steps.

Quick Sort (cont.)

- On the other hand, an unfortunate choice of the pivot could divide the array into two parts, one that contains no elements and another that contains $n-1$ elements.

n

n−2

n−1

n

- In this case, Quick Sort takes $O(n^2)$ steps.

Quick Sort (cont.)

- Various strategies are used to choose the pivot. (None is perfect.)

- Pick the first element (worst-case scenario is a nearly-sorted or nearly-inverse-sorted array).

- Take the median of the first, last, and middle elements. This is often used in practice, since it behaves well on the nearly-sorted case, which can be quite common in some applications.