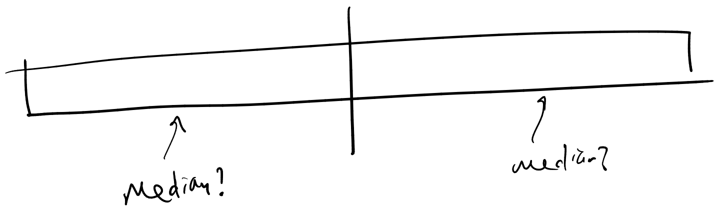
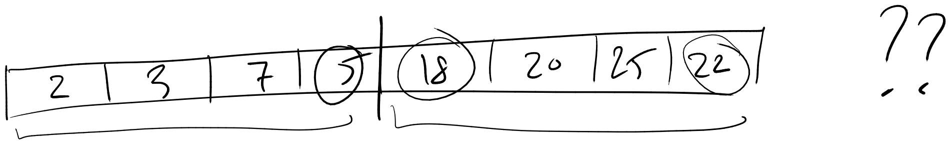
Problem: given an array with $n$ integers.
Find the median value in the array.

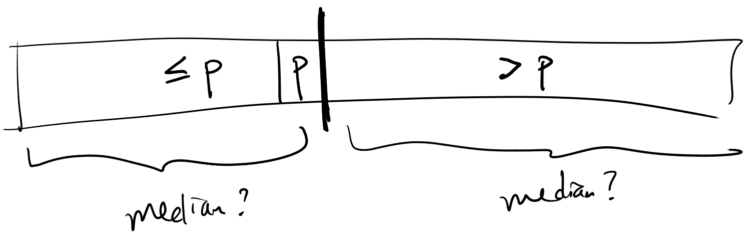
Approach 1: sort, then look up index $\lfloor n/2 \rfloor$ takes $\Theta(n \lg n)$

Let's consider divide + conquer?




Other ways to split input?

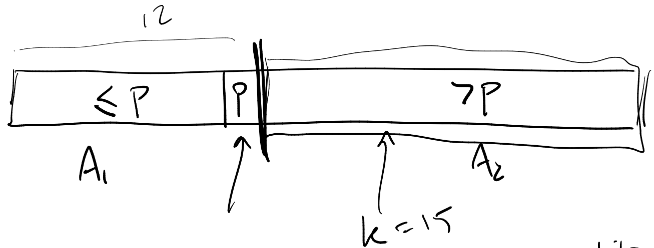



partition — $\Theta(n)$.

Feels like we're stuck — finding median of both sides seems like it doesn't help us find overall median.

We are going to fix this by solving a harder (more general) problem.
"Generalizing the induction hypothesis / recursion":

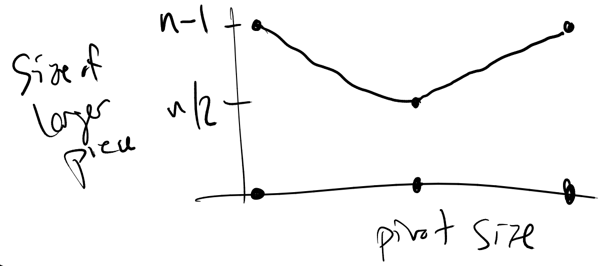
We will solve the more general problem of finding the "$k^{th}$ smallest" — i.e. find the element that would be @ index $k$ in the sorted array.




precondition: $0 \le k < |A|$

find $k^{th}$ smallest item in A, ie find sort(A)[k].

```
QuickSelect(A, k):
    if |A| = 1:
        return A[0].
    else:
        pick pivot p randomly.
        A1, A2 <- partition(A, p)
        if k < |A1|:
            return QuickSelect(A1, k)
        else:
            return QuickSelect(A2, k - |A1|).
```




on average, biggest piece is 3/4 n.

$Q(n)$ = time on array of size n.

partition.

$$Q(1) = 1$$
$$Q(n) = 1 \cdot Q(n/(4/3)) + O(n)$$

Use Master Theorem!

$a = 1, \quad b = 4/3, \quad d = 1.$

$1 < (4/3)^1$, Case 1 of Master Thm.

Hence overall, this takes $O(n)$ time!

If $0 \le k < |A|$,

**Theorem.** $\text{QUICKSELECT}(A, k) = \text{sort}(A)[k]$.

**Proof** by (strong) induction on $|A|$.

- Base case: when $|A| = 1$, $0 \le k < |A|$ then $k$ must be $0$. Hence $\text{sort}(A)[k]$

$$= A[0] = \text{QUICKSELECT}(A, k).$$

- Induction step:

Let $n > 1$ and suppose that

$$\text{QUICKSELECT}(A, k) = \text{sort}(A)[k]$$

whenever $|A| < n$. Then we must show it works on arrays of size $n$.

etc.