Selection

**NOTE:**
- Min = 16
- is greater than 
  Q1 - 1.5(Q3 - Q1) 
  = 18.5 - 1.5(2) 
  = 15.5 
  SO...stop at Min

- Max = 23
- is less than 
  Q3 + 1.5(Q3 - Q1) 
  = 20.5 + 1.5(2) 
  = 23.5 
  So...stop at Max.
Selection given a set of (distinct) elements, finding the element larger than \( i - 1 \) other elements

Selection with...

\( i=n \) is finding maximum
\( i=1 \) is finding minimum
\( i=n/2 \) is finding median
Maximum

Selection for any i is $O(n)$ runtime

Find max in $O(n)$?
Maximum

Selection for any $i$ is $O(n)$ runtime

Find max in $O(n)$?

$max = A[1]$
for $i = 2$ to $A$.length
    if ($A[i] > max$)
        $max = A[i]$
Max and min

It takes about $n$ comparisons to find max

How many would it take to find both max and min at same time?
Max and min

It takes about $n$ comparisons to find max

How many would it take to find both max and min at same time?

Naïve = $2n$
Smarter = $3/2 n$
Max and min

smin = min(A[1], A[2])
smax = max(A[1], A[2])
for i = 3 to A.length step 2
    if (A[i] > A[i+1])
        smax = max(A[i], smax)
        smin = min(A[i+1], smin)
    else
        smax = max(A[i+1], smax)
        smin = min(A[i], smin)
Randomized selection

Remember quicksort?

Partition step

Choose 31 as the pivot

Recursively sort subsequence on each side of pivot
Randomized selection

To select i:

1. Partition on random element

2. If partitioned element i, end otherwise recursively partition on side with i
Randomized selection

{2, 6, 4, 7, 8, 4, 7, 2} find i = 5
Randomized selection

\{2, 6, 4, 7, 8, 4, 7, 2\} find i = 5

Pick pivot = 4

\{2, 6, 4, 7, 8, 2, 7, 4\}
Randomized selection

\{2, 4, 6, 7, 8, 2, 7, 4\}
\{2, 4, 2, 7, 8, 6, 7, 4\}
\{2, 4, 2, 7, 8, 6, 7, 4\}
\{2, 4, 2, 4, 7, 8, 6, 7\}
1, 2, 3, 4, 5, 6, 7, 8

i=5 on green side, recurse
Randomized selection

\{7, 8, 6, 7\} pick pivot = 6
\{7, 8, 7, 6\}
\{7, 8, 7, 6\}
\{7, 8, 7, 6\}
\{7, 8, 7, 6\}
\{6, 7, 8, 7\}

5, 6, 7, 8

found i=5, value = 6
Randomized selection

Quicksort runs in $O(n \lg n)$, but we only have sort one side and sometimes stop early.

This gives randomized selection $O(n)$ running time (proof in book, I punt)
Randomized selection

Just like quicksort, the worst case running time is $O(n^2)$

This happens when you want to find the min, but always partition on the max
A worst case $O(n)$ selection is given by Select: (see code)
1. Make $n/5$ groups of 5 and find their medians (via sorting)
2. Recursively find the median of the $n/5$ medians (using Select)
3. Partition on median of medians
4. Recursively Select correct side
Proof of the general case:

\[ T(n) = \sum_i T(k_i \cdot n + q_i) + O(n) \]

// assume \( T(n) \) is \( O(n) \)

\[ T(n) = c \cdot n - c \cdot n + c \sum_i (k_i \cdot n + q_i) + a \cdot n \]

If \( T(n) \) is \( O(n) \) then...

\[ -c \cdot n + c \sum_i (k_i \cdot n + q_i) + a \cdot n \leq 0 \]

\[ a \cdot n \leq c(n(1 - \sum_i k_i) - \sum_i q_i) \]
Select

\[
a \cdot n \leq c(n(1 - \sum_i k_i) - \sum_i q_i)
\]

\[
\frac{a \cdot n}{n(1 - \sum_i k_i) - \sum_i q_i} \leq c
\]

// Pick \( n > 2(\text{sum}_i q_i/(1 - \text{sum}_i k_i)) \)

\[
\frac{2a(\sum q_i/(1 - \sum k_i))}{\sum_i q_i} \leq c
\]

\[
\frac{2a}{1 - \sum_i k_i} \leq c
\]

Done as \( \sum_i k_i < 1 \) (just need show for this \( n, O(1) \))
Select

Select runs in:
\[ T(n) = T(\text{ceiling}(n/5)) + T(7n/10 + 6) + O(n) \]

By the previous proof this is \( O(n) \):
\[ \text{ceiling}(n/5) + 7n/10 + 6 \leq n/5 + 1 + 7n/10 + 6 = 9n/10 + 7 \]
\[ \sum_{i} k_i = 9/10 < 1 \text{, done} \]
Does this work for making:

(1) \(n/3\) groups of 3?
(2) \(n/7\) groups of 7?
(3) \(n/9\) groups of 9?