Outline

- K\textsuperscript{th} order statistics
  - Min
  - Min/Max
- Proof of upper and lower bounds
- Adversary argument

\textbf{K-th Order Statistics}

- Input: Set of integers S
- Output: k\textsuperscript{-th} smallest integer in S

What if k=1?
What if k=n?
What if k=n/2?

\textbf{FIND-MIN}

- How many comparisons does it take to find the minimum in a set of integers?
- Answer: n-1

\textbf{In worst case}

\textbf{Upper Bound for FIND-MIN}

Upper Bound Theorem: Finding the minimum in a set of n integers requires no more than n-1 comparisons.

Proof: Give algorithm

\textbf{Lower Bound for FIND-MIN}

Lower Bound Theorem: Finding the minimum in a set of integers requires at least n-1 comparisons.
Proof of Lower Bound:
• Consider an algorithm A on input of size n.
• Let G be a graph with a vertex for each input integer. Initially G has no edges. When A compares two input values, we’ll add an edge between the corresponding vertices of G.
• A cannot conclude until G has _______ edges.
• Thus A cannot conclude until it has made ______ comparisons.

Upper Bound for FIND-MIN/MAX
• Upper Bound Theorem: Finding the minimum and maximum in a set of n integers requires no more than \[\lceil \frac{3n}{2} \rceil - 2\] comparisons.
• Proof: Give an algorithm

Proof of Upper Bound:
• Algorithm for even n:
  – Make n/2 pair-wise comparisons
  – Find the maximum of the winners with n/2-1 comparisons
  – Find the minimum of the losers with n/2-1 comparisons
• Algorithm for odd n:
  – Run even algorithm on first n-1 integers
  – Compare the min and max to the last integer

Lower Bound for FIND-MIN/MAX
• Lower Bound Theorem: Finding the minimum and maximum in a set of n integers requires at least \[\lceil \frac{3n}{2} \rceil - 2\] comparisons.
• Proof: Adversary argument

Example of an adversary
You pick a number y between 1 and 100
I have to guess y by posing queries of the form “Is it \(y\)?”
You answer “yes, \(x=y\)” or “no, \(y<x\)” or “no, \(y>x\)”
• How many queries can you force me to make?
• Prove it!

Adversary Argument to prove bound B
(for FIND MIN/MAX)

Find-Min/Max Algorithm
\(\text{A}\)

Adversary Algorithm
\(\text{A}\)

set of integers S:
A makes at least B comparisons on input S
FIND-MIN/MAX Adversary - Accounting
- Adversary = interactive comparison oracle
- Accounting scheme: For x in S
  \[ b_{\text{MAX}}(x) = \begin{cases} 1 & \text{if the algorithm can rule out } x \text{ as the largest integer} \\ 0 & \text{otherwise} \end{cases} \]
  \[ b_{\text{MIN}}(x) = \begin{cases} 1 & \text{if the algorithm can rule out } x \text{ as the smallest integer} \\ 0 & \text{otherwise} \end{cases} \]

FIND-MIN/MAX Adversary - Strategy
- On query “Is x < y?”
- Answer consistently with previous answers
- If yes and no both consistent then answer so as to minimize the changes in \( b_{\text{MAX}} \) and \( b_{\text{MIN}} \) variables

FIND-MIN/MAX Adversary - Analysis
Consider a query “Is x < y?”
- If NO: \( b_{\text{MIN}}(x) \rightarrow 1 \) and \( b_{\text{MAX}}(y) \rightarrow 1 \)
- If YES: \( b_{\text{MAX}}(x) \rightarrow 1 \) and \( b_{\text{MIN}}(y) \rightarrow 1 \)

Proof of Lower Bound:
- Claim: At most \( \lfloor n/2 \rfloor \) queries can result in the change of two \( b_{\text{MIN/MAX}} \) variables
- Claim: 2n-2 changes must occur before the algorithm concludes
  \[ \implies \lfloor n/2 \rfloor + (2n-2) - 2\lfloor n/2 \rfloor \text{ queries are necessary} \]

Double 0’s
- Input: n-bit vector of 0/1’s
- Question: are there two adjacent 0’s
- How many queries are needed?

Exercise
- What is a good adversary strategy?
- What is a good algorithm strategy
Upper Bound

Claim: Double 0’s can be solved with $f(n)$ queries where:
$$f(n) = \begin{cases} n-1 & \text{if } n \equiv 1 \mod 3 \\ n & \text{otherwise} \end{cases}$$

Lower Bound

Double 0’s cannot be solved with fewer than $g(n)$ queries where:
$$g(n) = \begin{cases} n-1 & \text{if } n \equiv 1 \mod 3 \\ n & \text{otherwise} \end{cases}$$

Proof

• Do as homework!