Lower bounds for Selection

A bound on the complexity of a problem is a bound on 
any algorithm that solves the problem

Technique:
Adversary Argument: Evil oracle dynamically 
decides outcomes of comparisons to thwart any 
algorithm.

See handout on Adversary arguments.

Lower bounds for Selection (cont)

Model of Computation: Tournaments

Keys (distinct values) \(\overset{\rightarrow}{\text{Players}}\)
Comparisons \(\overset{\rightarrow}{\text{ Matches}}\)
Algorithms \(\overset{\rightarrow}{\text{ Schedule of matches}}\)
Running time \(\overset{\rightarrow}{\text{ Number of matches}}\)

Assumption: Outcomes of matches is consistent

Lower bounds for Selection (cont)

Problems | Complexity | \(V_k(n)\) | \(U_k(n)\) | \(W_k(n)\)
--- | --- | --- | --- | ---
Find the \(k\)th player out of \(n\) | \(V_k(n)\) | | | |
Find the top \(k\) players out of \(n\) | \(U_k(n)\) | | | |
Find the 1st, 2nd, …, \(k\)th players out of \(n\) | \(W_k(n)\) | | | |

\(U_k(n) \leq V_k(n) \leq W_k(n)\)

Proof: Play a single elimination tournament.

All players except champion lose once \(\Rightarrow n-1\) matching.

So \(V_k(n) \leq n-1\)
Lower bounds for Selection (cont)

\( V_1(n) = V_n(n) = n - 1 \)

**Proof:** (continued) To show that \( V_1(n) \geq n - 1 \)
suppose you have played fewer than \( n-1 \) matches.
Then there are at least two players who have never lost and either could be the top player. So the top player can not be identified with fewer than \( n-1 \) matches.

\[ V_1(n) = n - 1 \quad \text{QED} \]

Lower bounds for Selection (cont)

\( W_2(n) = n + \left[ \lg n \right] - 2 \)

**Proof:** Play a single elimination tournament among the \( n \) players with as many matches as possible in each round. Uses \( n-1 \) matches.

\[ n = 7 \]

Lower bounds for Selection (cont)

\( V_2(n) = W_2(n) \)

**Proof:** \( V_2(n) \leq W_2(n) \) since if you know the 1st and 2nd players than you know the 2nd player.
To show \( V_2(n) \geq W_2(n) \) assume we know player B is 2nd. Since player B is not first it has lost a match, say to player A. If any player is better than A then B could not be second. So A must be the top player. QED

It is possible for \( U_2(n) < V_2(n) = W_2(n) \).

Lower bounds for Selection (cont)

\( W_2(n) = n + \left[ \lg n \right] - 2 \)

**Proof:** (continued) If the champion is A, collect all of the players that lost to A into a set \( S \). Play a second single elimination tournament among the players in \( S \). This requires \( |S|-1 \) matches.

\[ S = \{ B, C, D \} \]
**Lower bounds for Selection (cont)**

\[ W_2(n) = n + \lceil \lg n \rceil - 2 \]

**Proof:** (continued) The champion A is the best player since everyone else has lost a match.

The players not in S, cannot be second since they have lost to a player who is not the best player.

Of the players in S, only the champion of the second tournament has not lost twice.

So the champion of the second tournament is the number 2 player.

---

**Lower bounds for Selection (cont)**

\[ W_2(n) = n + \lceil \lg n \rceil - 2 \]

**Proof:** (continued) The number of players in S is the number of matches played by A.

This is at most the height of the tree of the first tournament.

This tree has \( n \) leaves and height at most \( \lceil \lg n \rceil \).

The total number of matches played is

\[ n - 1 + |S| - 1 = n + \lceil \lg n \rceil - 2 \]

So \( W_2(n) \leq n + \lceil \lg n \rceil - 2 \)

---

**Lower bounds for Selection (cont)**

\[ W_2(n) = n + \lceil \lg n \rceil - 2 \]

**Proof:** (continued) To show \( W_2(n) \geq n + \lceil \lg n \rceil - 2 \)

we need to devise an oracle which will force this number of matches regardless of how the matches are organized.

Notation: \( TOP = \{ \text{currently undefeated players} \} \)

\( \text{Win}(x) = \# \text{ of matches } x \text{ has played and won against players that were (previously) undefeated} \)

---

**Lower bounds for Selection (cont)**

\[ W_2(n) = n + \lceil \lg n \rceil - 2 \]

**Proof:** (continued)

\[ DOM(x, 0) = \{ x \} \]

\[ DOM(x, i) = \{ y \mid y \text{ 's first defeat was to a player in } DOM(x, i - 1) \} \]

\[ DOM(x) = \bigcup_{i=0}^{n} DOM(x, i) \]

At start:

\[ \forall x \ DOM(x) = \{ x \} \]

\[ TOP = \{ \text{all players} \} \]

\[ \forall x \ \text{Win}(x) = 0 \]
Claim: After \( m \) matches are played, if \( x \) is still in \( \text{TOP} \), then

\[
\begin{align*}
\text{Oracle decides matches as follows:} & \\
x \text{ wins if } x, y & \in \text{TOP} \text{ and } \text{Win}(x) \geq \text{Win}(y) \\
\text{or if } x & \in \text{TOP} \text{ and } y \notin \text{TOP} \\
\text{else whatever is consistent}
\end{align*}
\]

Proof (continued)

By induction on \( m \).
Base: \( m=0 \)
\[
|\text{DOM}(x)| = |\{x\}| = 1 = 2^0 = 2^{\text{Win}(x)}
\]

\[
\begin{align*}
\text{Step: } m>0. \text{ Assume claim holds after } m-1 \text{ matches.} \\
\text{Case 1: } x \text{ does not play in match } m. & \\
\text{Then } \text{DOM}(x) \text{ and } \text{Win}(x) \text{ don’t change.} \\
\text{Case 2: } \text{In match } m, \ x \text{ plays } y \notin \text{TOP}. & \\
\text{Then } \text{DOM}(x) \text{ and } \text{Win}(x) \text{ don’t change.} \\
\text{Case 3: } \text{In match } m, \ x \text{ plays } y \in \text{TOP}. & \\
\text{newDOM}(x) \leftarrow \text{DOM}(x) \cup \text{DOM}(y) \\
\text{and newWin}(x) \leftarrow \text{Win}(x)+1 .
\end{align*}
\]

\[
|\text{newDOM}(x)| \leq 2^{\text{Win}(x)} + 2^{\text{Win}(y)} \leq 2^{\text{Win}(x)+1} = 2^{\text{newWin}(x)}
\]

Results:

\[
\begin{align*}
V_2(n) &= n + \lceil \lg n \rceil - 2 = 5 + \lceil \lg 5 \rceil - 2 = 6 \\
U_2(5) &= 5
\end{align*}
\]

It’s possible to know the top 2 players out 5 without knowing the top player.

These two players are both better than 3 others and so are 1st or 2nd.
Lower bounds for Selection (cont)

Finding first and last player requires at least \( \left\lceil \frac{3n}{2} \right\rceil - 2 \) matches.

See handout.