**Theorem 1.** *Finding both the minimum and maximum elements in an unsorted array requires at least* $\lceil \frac{3n}{2} \rceil - 2$ *comparisons in the worst case.*

*Proof.* Here only the situation when $n$ is even will be discussed. When $n$ is odd, the analysis is of little difference.

Let $A$ be the set of all $n$ elements in an array. We define three auxiliary sets which are initially empty, nominally *Big*, *Small* and *Discarded*.

Given any algorithm *Alg*, let $c_i$ denote *Alg*'s $i$-th comparison action. Note that *Alg* will performance a sequence of comparisons $c_1, \ldots, c_m$ in order to find out both the maximum and minimum elements. Our task is to show that $m \geq \frac{3}{2}n - 2$.

Consider each comparison $c_i$, which shows the relationship $a_x \geq a_y$ for some elements $a_x$, $a_y$. Define operations on the four sets $A$, *Big*, *Small* and *Discarded* as following:

- If $a_x \in A$, move $a_x$ to *Big*.
- If $a_y \in A$, move $a_y$ to *Small*.
- If $a_x \in$ *Small*, move $a_x$ to *Discarded*.
- If $a_y \in$ *Big*, move $a_y$ to *Discarded*.
- Otherwise, keep $a_x$ (or $a_y$) unmoved.

We claim but will not prove the following properties:

- Elements in *Big* cannot be minimum. Elements in *Small* cannot be maximum. And elements in *Discarded* can be neither minimum nor maximum.
- Any two elements in the same set $X$, where $X \in \{A, Big, Small\}$, are not compared yet.
- To find out both the minimum and maximum. Finally $A$ must be empty. *Big* and *Small* must have only one element each.

To do worst case analysis, suppose there is an evil adversary *Adv*. *Adv* controls the input data and it always tries to give the 'worst' (i.e. of least information) comparison result while ensuring consistency. Given two elements $a_x$ and $a_y$, we define *Adv*'s behavior as following :

1. If $a_x, a_y \in X$ where $X \in \{A, Big, Small\}$, report $a_x \geq a_y$.
2. If $a_x \in$ *Big* and $a_y \in A$, report $a_x \geq a_y$ (vise versa).
3. If $a_x \in A$ and $a_y \in$ *Small*, report $a_x \geq a_y$ (vise versa).
4. If $a_x \in$ *Big* and $a_y \in$ *Small*, report $a_x \geq a_y$ (vise versa).
5. Otherwise, it must be $a_x, a_y \in$ *Discarded*. Depending on whether $a_x$, $a_y$ were compared, report a consistent result.

Not difficult to see that *Adv*'s reports are consistent.

Note that for each comparison either (i) at most two elements are moved from $A$ to *Big*/*Small* or (ii) at most one element is moved from *Big*/*Small* to *Discarded*. Thus to satisfy the final requirement that $A$ is empty and *Big* and *Small* have one element each, it takes at least $n/2$ steps to move out all elements from $A$ according to (i), and it takes at least $n-2$ steps to move out extra elements from *Big*/*Small* according to (ii). Then the total number of comparisons needed is at least $n/2 + n - 2 = \frac{3}{2}n - 2$. □