Selection Sort

A simple $O(n^2)$ sorting algorithm is selection sort.

Sweep through all the elements to find the smallest item, then the smallest remaining item, etc. until the array is sorted.

```
Selection-sort(A)
        for i = 1 to n
                for j = i + 1 to n
                        if (A[j] < A[i]) then swap(A[i],A[j])
```

It is clear this algorithm must be correct from an inductive argument, since the ith element is in its correct position.

It is clear that this algorithm takes $O(n^2)$ time.

It is clear that the analysis of this algorithm cannot be improved because there will be $n/2$ iterations which will require at least $n/2$ comparisons each, so at least $n^2/4$ comparisons will be made. More careful analysis doubles this.

Thus selection sort runs in $\Theta(n^2)$ time.