1 Bubble Sort

1.1 Pseudocode

Algorithm 1 Bubble Sort

```
for i = n  down to  2 do
    for j = 1  to i − 1 do
        if A[j] > A[j + 1] then
            A[j] ↔ A[j + 1]
        end if
    end for
end for
```

1.2 Analysis of Comparisons

Bubble Sort is designed in such a manner such that the state of the array to be listed does *not* change the number of comparisons it makes.

$$\begin{aligned}\sum_{i=2}^{n}\sum_{j=1}^{i-1} 1 &= \sum_{i=2}^{n} i-1 \\ &= \sum_{i=1}^{n-1} i \\ &= \frac{(n-1)n}{2} = \binom{n}{2}\end{aligned}$$

1.3 Analysis of Exchanges

Worst Case

Worst case is when the list is reverse sorted, there will be the same number of exchanges as comparisons.

$$\frac{(n-1)n}{2}$$

Best Case

Best case is when the list is already sorted, in which there will be zero exchanges.

Average Case

To find the average case we must count the transpositions (two elements that are out of order related to one another). In best case there are no transpositions, and in worst case there are $\frac{(n-1)n}{2}$ transpositions. In a randomly permuated array each element is equally likely to be out of order so the total number of average case exchanges is half the comparisons.

$$\frac{1}{2} \cdot \frac{(n-1)n}{2} = \frac{(n-1)n}{4}$$