

### Elementary Sorts

1. Which of the three elementary methods (selection, insertion and bubble sort) do you expect to run fastest for an array in reverse-order? Why?

(Hint: You can solve this problem by comparing timings for the programs, but this exercise requires no coding. The recommend approach is to count the number of comparisons and exchanges each of the three methods would require to complete the sort).

2. Implement a version of bubble sort that alternates left-to-right and right-to-left passes through the data. This (faster but more complicated) algorithm is called *shaker sort*.

(Remember to test your program!).

### Shellsort

3. For Shellsort, consider the two increment sequences below:

a) 1, 3, 7, 15, 31, 63, ... ( $2^i - 1$ )

b) 1, 19, 209, 505, ... ( $9 \cdot 4^i - 9 \cdot 2^i + 1$ )

Calculate the number of comparisons used by each sequence when:

- i) Sorting 10,000 random numbers.
- ii) Sorting an array that is already in order.
- iii) Sorting an array in reverse order.