**2.1.1**

**0 1 2 3 4 5 6 7 8 9 10 11**

**E A S Y Q U E S T I O N**

**A E S Y Q U E S T I O N**

**A E E Y Q U S S T I O N**

**A E E I Q U S S T Y O N**

**A E E I N U S S T Y O Q**

**A E E I N O S S T Y U Q**

**A E E I N O Q S T Y U S**

**A E E I N O Q S T Y U S**

**A E E I N O Q S S Y U T**

**A E E I N O Q S S T U Y**

**A E E I N O Q S S T U Y**

**A E E I N O Q S S T U Y**

**2.1.2** What is the maximum number of exchanges involving any particular item during selection sort? What is the average number of exchanges involving an item?

**If the largest element is originally in the first position and all elements to the right are already in order, the first element will be exchanged N times.**

**Not sure how to do the calculation but it has these components:**

* **Some elements will by chance already be in the correct position, these will not need to be exchanged. The second S above is an example.**
* **Each element that is in the wrong position will be exchanged at least once.**
* **Some elements may be exchanged more than once depending on the arrangement of remaining elements after the first exchange.**

**2.1.3** Give an example of an array of N items that maximizes the number of times the test a[j] < a[min] succeeds (and, therefore, min gets updated) during the operation of selection sort ([ALGORITHM 2.1](ms-local-stream://EpubReader_A7E0398FB7E648908B1F59A68DAC6346/Content/OEBPS/html/ch02.html#ch02sb05)).

If the array is already sorted in reverse order:

**z y x w v u t n. fails = 6**

**t y x w v u z n. fails = 4**

**t u x w v y z n. fails = 2**

**t u v w x y z n. fails = 0 – sorted, total = 12**

**If the array has duplicates and consists of two reversed-order sub arrays:**

**z y x w y x w**

**z y x w y x w n. updates = 3**

**w y x z y x w n. updates = 2**

**w w x z y x y n. updates = 0**

**w w x z y x y n. updates = 2**

**w w x x y z y n. updates = 0**

**w w x x y z y n. updates = 1**

**w w x x y y z n. updates = 0 – sorted, total = 8**

**2.1.4** Show, in the style of the example trace with [ALGORITHM 2.2](ms-local-stream://EpubReader_A7E0398FB7E648908B1F59A68DAC6346/Content/OEBPS/html/ch02.html#ch02sb07), how insertion sort sorts the array E A S Y Q U E S T I O N.

E A S Y Q U E S T I O N start

A E S Y Q U E S T I O N i = 1; j = 0

A E S Y Q U E S T I O N i = 2; j = 2

A E S Y Q U E S T I O N i = 3; j = 2

A E S Y Q U E S T I O N i = 4; j = 2

A E Q S Y U E S T I O N i = 5; j = 4

A E Q S U Y E S T I O N i = 6; j = 2

A E E Q S U Y S T I O N i = 7; j = 5

A E E Q S S U Y T I O N i = 8; j = 6

A E E Q S S T U Y I O N i = 9; j = 3

A E E I Q S S T U Y O N i = 10; j = 4

A E E I O Q S S T U Y N i = 11; j = 4

**2.1.5** For each of the two conditions in the inner for loop in insertion sort ([ALGORITHM 2.2](ms-local-stream://EpubReader_A7E0398FB7E648908B1F59A68DAC6346/Content/OEBPS/html/ch02.html#ch02sb07)), describe an array of N items where that condition is always false when the loop terminates.

**2.1.6** Which method runs faster for an array with all keys identical, selection sort or insertion sort?

**2.1.7** Which method runs faster for an array in reverse order, selection sort or insertion sort?

Selection sort has to do the same number of compares, no matter the order: N \* N / 2. If original is in reverse order, each step of selection sort also correctly sorts the end of the array, so once the sort index reaches the middle element, the array is fully sorted and there are no more exchanges.

Insertion sort will have to do *i* compares each step: N \* N / 2 and move *i* elements per step: N \* N / 2.

Selection: quadratic compares, linear exchanges

Insertion: quadratic compares, quadratic exchanges

Insertion will be slower for reverse-ordered arrays.

**2.1.8** Suppose that we use insertion sort on a randomly ordered array where items have only one of three values. Is the running time linear, quadratic, or something in between?

**2.1.9** Show, in the style of the example trace with [ALGORITHM 2.3](ms-local-stream://EpubReader_A7E0398FB7E648908B1F59A68DAC6346/Content/OEBPS/html/ch02.html#ch02sb11), how shellsort sorts the array E A S Y S H E L L S O R T Q U E S T I O N.

**2.1.10** Why not use selection sort for h-sorting in shellsort?

**2.1.11** Implement a version of shellsort that keeps the increment sequence in an array, rather than computing it.

**2.1.12** Instrument shellsort to print the number of compares divided by the array size for each increment. Write a test client that tests the hypothesis that this number is a small constant, by sorting arrays of random Double values, using array sizes that are increasing powers of 10, starting at 100.

**Creative Problems**

**2.1.13** *Deck sort.* Explain how you would put a deck of cards in order by suit (in the order spades,