Foundations of Computer Science (Semester 2) – 2015

Assessed Exercise Sheet 6 – 10% of Continuous Assessment Mark

Deadline : 11pm Sunday 1st March, via Canvas

Question 1 (16 marks)

Shaker Sort (also called Bidirectional Bubble Sort) successively compares adjacent pairs of items in an array, and exchanges them if they are in the wrong order. It alternatively passes forwards and backwards through the array until no more exchanges are needed. Explain why this algorithm is guaranteed to terminate with the array sorted.

Work through the sorting of array [3, 2, 4, 5, 1, 6] using this algorithm, writing down the direction of the pass at each swap, the swaps that occur, and the array after each swap.

Question 2 (16 marks)

The following code will sort an array `a` of finite size `n`:

```
for( i = 1 ; i < n ; i++ ) {
    j = i;
    c = a[j];
    while( j > 0 && c < a[j-1] ) {
        a[j] = a[j-1];
        j--;
    }
    a[j] = c;
}
```

Which sorting algorithm discussed in the lectures is this?

Explain why it is guaranteed to terminate with the array `a` sorted.

Work through the sorting of array [5, 3, 7, 8, 1, 9] using this algorithm, writing down the values of `i, j, c` and the array `a` at the end of each iteration of the `for` loop.

Question 3 (16 marks)

The following code will also sort an array `a` of finite size `n`:

```
for( i = 0 ; i < n-1 ; i++ ) {
    k = i;
    for( j = i+1 ; j < n ; j++ ) {
        if( a[j] < a[k] ) k = j;
    }
    c = a[i];
    a[i] = a[k];
    a[k] = c;
}
```

Which sorting algorithm discussed in the lectures is this?

Explain why it is guaranteed to terminate with the array `a` sorted.

Work through the sorting of array [5, 3, 7, 8, 1, 9] using this algorithm, writing down the values of `i, j, k` and the array `a` at the end of each iteration of the `i for` loop.

Question 4 (18 marks)

One often needs to sort items which might have identical keys (e.g., ages in years) in a way that keeps items with identical keys in their original order (e.g., alphabetical). So, if we denote the original order of an array of items by subscripts, we want the subscripts to end up in order for each set of items with identical keys. For example, if we start out with the array $[5_1, 4_2, 6_3, 5_4, 6_5, 7_6, 5_7, 2_8, 9_9]$ it should be sorted to $[2_8, 4_2, 5_1, 5_4, 5_7, 6_3, 6_5, 7_6, 9_9]$ and not to $[2_8, 4_2, 5_4, 5_1, 5_7, 6_3, 6_5, 7_6, 9_9]$. Sorting algorithms which do this are said to be *stable*. For each of Bubble Sort, Insertion Sort and Selection Sort, say whether that algorithm is stable, and explain why. For cases where the algorithm is not stable, give a simple example that illustrates why.

Question 5 (18 marks)

In the lectures notes (section 6.4), a procedure `insert(v,bst)` is presented that inserts a value `v` into a binary search tree `bst`. Summarize, using a list of possible cases that need to be dealt with, how that procedure works.

Describe the simplest possible ways the procedure could be modified to accommodate duplicate entries.

Comment on how the possible modifications would affect the stability of a Treesort algorithm based on it.

Question 6 (16 marks)

Show how the array [21, 26, 23, 10, 18, 35, 13, 4, 16, 6, 25, 27, 33, 39] would be sorted if the Treesort algorithm was used. Make it clear what is being done at each stage, and show the data structure at the key stages.