CS1713 – ALGORITHM DESIGN AND ANALYSIS

Shellsort Pseudocode

ALGORITHM Shellsort(A[0…n-1])

```
// Sorts array A[0….n-1] by a succession of h-sorts
// Different sequences of h create different runtimes
// This code uses the sequence 1, 4, 13, 40, 121, 364 ….. in reverse

// First find start value for h
h = 1
while (h <= (n – 1)/9)
     h = 3*h + 1

// Now carry out the sorts
while (h > 0)
  // For a fixed value of h the array is sorted multiple times
  // using insertion sort of every hth item. This is called h-sort
  for i = h to n – 1
     v = A[i]
     j = i – h
     while (j >= 0) and (A[j] > v)
         A[j + h] = A[j]
         j = j – h
     A[j + h] = v
   h = ⌊h/3⌋
```

Notes:

1. Shellsort is a generalized version of Insertion Sort. Compare with the book's Insertion Sort pseudocode (P. 134) - the **for loop** is exactly the same except that h replaces 1 (because it is an h-sort).

2. The sequence for this Shellsort is actually $(3^k – 1)/2$ for k = 1, 2, 3, 4, 5……..

3. Different sequences of h can be used to improve the runtime slightly. The sequence is called the **gap sequence**.

4. The run-time is hard to analyze but for the algorithm above it is $O(n^{3/2})$. Better gap sequences can give slightly better runtimes.