UCS1302
DATA STRUCTURES

Shell sort

Session Objectives

- To learn about Shell sort algorithm

Session Outcomes

- At the end of this session, participants will be able to
  - Sort the numbers using shell sort

Shell sort

Dr. B. Bharathi
SSNCE

September 19, 2019

Shellsort

- Invented by Donald Shell in 1959.
- 1st algorithm to break the quadratic time barrier but few years later, a sub quadratic time bound was proven
- Shellsort works by comparing elements that are distant rather than adjacent elements in an array.

Shellsort

- Shellsort uses a sequence h_1, h_2, …, h_t called the ***increment sequence***. Any increment sequence is fine as long as $h_1 = 1$ and some other choices are better than others.
- Shellsort makes multiple passes through a list and sorts a number of equally sized sets using the insertion sort.
- Shellsort improves on the efficiency of insertion sort by *quickly* shifting values to their destination.

Shellsort

- Shellsort is also known as ***diminishing increment sort***.
- The distance between comparisons decreases as the sorting algorithm runs until the last phase in which adjacent elements are compared.
- After each phase and some increment h_k, for every ***i***, we have a[***i***] ≤ a [***i*** + h_k] all elements spaced h_k apart are sorted.
- The file is said to be h_k – sorted.

Empirical Analysis of Shellsort (Advantage)

- Advantage of Shellsort is that its only efficient for medium size lists. For bigger lists, the algorithm is not the best choice. Fastest of all O(N^2) sorting algorithms.
- 5 times faster than the bubble sort and a little over twice as fast as the insertion sort, its closest competitor.

Shellsort Best Case

- Best Case: The best case in the shell sort is when the array is already sorted in the right order. The number of comparisons is less.

Shellsort Worst Case

- The running time of Shellsort depends on the choice of increment sequence.
- The problem with Shell's increments is that pairs of increments are not necessarily relatively prime and smaller increments can have little effect.

Shellsort Examples

Sort: 18 32 12 5 38 33 16 2

8 Numbers to be sorted, Shell's increment will be floor(n/2)

* floor(8/2) ➔ floor(4) = 4

increment 4: 1 2 3 4

18 32 12 5 38 33 16 2

Step 1) Only look at 18 and 38 and sort in order ;
18 and 38 stays at its current position because they are in order.

Step 2) Only look at 32 and 33 and sort in order ;
32 and 33 stays at its current position because they are in order.

Shellsort Examples

Sort: 18 32 12 5 38 33 16 2

8 Numbers to be sorted, Shell's increment will be floor(n/2)

* floor(8/2) ➔ floor(4) = 4

increment 4: 1 2 3 4

18 32 12 5 38 33 16 2

Step 3) Only look at 12 and 16 and sort in order ;
12 and 16 stays at its current position because they are in order.

Step 4) Only look at 5 and 2 and sort in order ;
2 and 5 need to be switched to be in order.

Shellsort Examples (con't)

Sort: 18 32 12 5 38 33 16 2

Resulting numbers after increment 4 pass:

18 32 12 2 38 33 16 5

* floor(4/2) ➔ floor(2) = 2

increment 2: 1 2

18 32 12 2 38 33 16 5

Step 1) Look at 18, 12, 38, 16 and sort them in their appropriate location:

12 38 16 2 18 33 38 5

Step 2) Look at 32, 2, 33, 5 and sort them in their appropriate location:

12 2 16 5 18 32 38 33

Shellsort Examples (con't)

Sort: 18 32 12 5 38 33 16 2

* floor(2/2) ➔ floor(1) = 1

increment 1 : 1

12 2 16 5 18 32 38 33

2 5 12 16 18 32 33 38

The last increment or phase of Shellsort is basically an Insertion Sort algorithm.