



* Assignment No - 13 *

* Title -: Implement the heap sort algorithm implemented in java demonstration.

* Objective :-

1. To Understand Concept of heap in data structure.
2. To Understand Concept of features of java language.

* Learning Objective :-

1. To Understand Concept of heap in Data structure.
2. To Understand Concept & features of java language.

* Outcome :-

- Input: Numbers or Random Data (Integers).
- Output: Sorted data by heap sort.

* Theory :-

Heap sort :-

comparison based sorted technique. It is similar to selection sort where we find the maximum element and place it at. we repeat process for all element.

* What is heap sort :-

A binary heap is complete binary tree compare items where are sorted



in a special order such that value in present node is greater or smaller than the value in its two children.

A sorting algorithm that works by first organising the data to be sorted into a special type of binary tree called heap. The heap itself has by definition the largest value at the top of all tree. So the heap sort algorithm must preserve the order.

* Heap sort algorithm for sorting in increment order.

1. Build a max heap from TIP data.
2. The largest item stored at root firstly, Replaced it with last item of heap followed by reducing size of heap by 1.
3. Shell sort above loop until size of heap greater than 1

* Shell sort.

Shell sort, also known as Shell's sort method is an in-place comparison sort, it can be seen as either a generalization of sorting by exchange (bubble sort) sorting by insertion (insertion sort).

Time Complexity of shell sort in worst case performance $O(n^2)$ [worst case]. And Best case performance depends on gap sequence.

Pseudo code for shell sort

Sort on array $a[0 \dots n-1]$.

gap = [701, 301, 132, 57, 23, 10, 4, 1].

foreach (gap in gaps)

{

for ($i = \text{gap}; i < n; i += 1$)

{

temp = $a[i]$;

for ($j = 1; j \geq \text{gap}$ and $a[j - \text{gap}] > \text{temp};$
 $j -= \text{gap}$).

{

$a[i] = a[j - \text{gap}]$

}

$a[j] = \text{temp};$

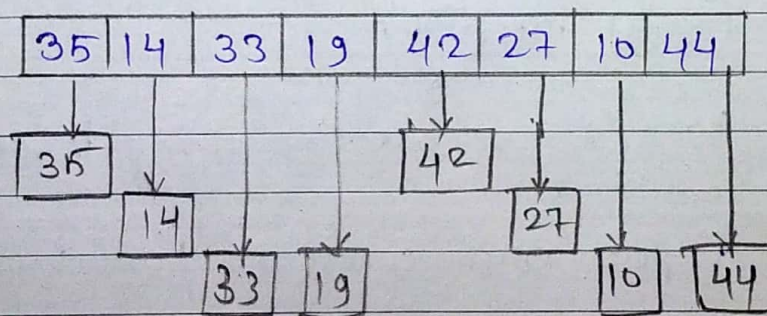
}

}

Ex-:

Values are:

[35, 14], [33, 19], [42, 27] and [10, 44].

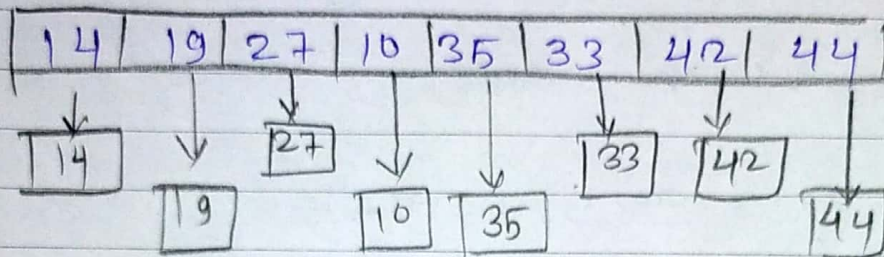


New sorted array :-

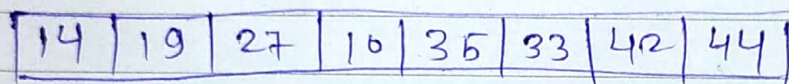
14 19 27 10 35 33 42 44



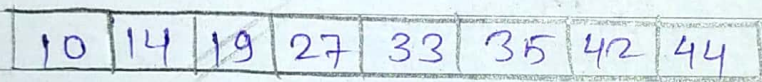
Then, we take interval of 2 and this gap generation two sub-list: [14, 27, 35, 42], [19, 10, 33, 44].



* then, compare and swap the values.



* Sorted Array:



* Algorithm for Shell sort:

- 1) Initialize the values of n .
- 2) Divide the list into smaller sub-list of Equal interval n .
- 3) Sort these sub-list using insertion sort.
- 4) Repeat until complete list is sorted.

* Conclusion:

Hence, we studied and implemented the heap & shell sort using java languages.