Shell Sort

> 💡 Shell sort is a generalized version of the insertion sort algorithm. It first sorts elements that are far apart from each other and successively reduces the interval between the elements to be sorted.

Algorithm Steps

1. Initialize the value of h.
2. Divide the list into smaller sub-list of equal interval h.
3. Sort these sub-lists using insertion sort.
4. Repeat until complete list is sorted.

The interval between the elements is reduced based on the sequence used. Some of the optimal sequences that can be used in the shell sort algorithm are:

- **Shell's original sequence:** `N/2 , N/4 , …, 1`
- **Knuth's increments:** `1, 4, 13, …, (3k – 1) / 2`
- **Sedgewick's increments:** `1, 8, 23, 77, 281, 1073, 4193, 16577...4j+1+ 3·2j+ 1`
- **Hibbard's increments:** `1, 3, 7, 15, 31, 63, 127, 255, 511…`

- **Papernov & Stasevich increment:** `1, 3, 5, 9, 17, 33, 65,...`
- **Pratt:** `1, 2, 3, 4, 6, 9, 8, 12, 18, 27, 16, 24, 36, 54, 81....`

1. Suppose, we need to sort the following array.

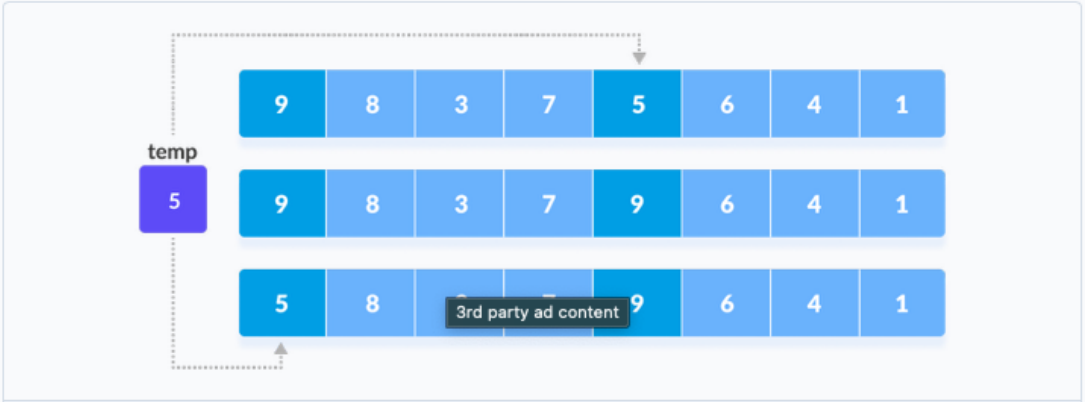
Initial array

2. We are using the shell's original sequence (`N/2, N/4, ...1`) as intervals in our algorithm.

   In the first loop, if the array size is `N = 8` then, the elements lying at the interval of `N/2 = 4` are compared and swapped if they are not in order.

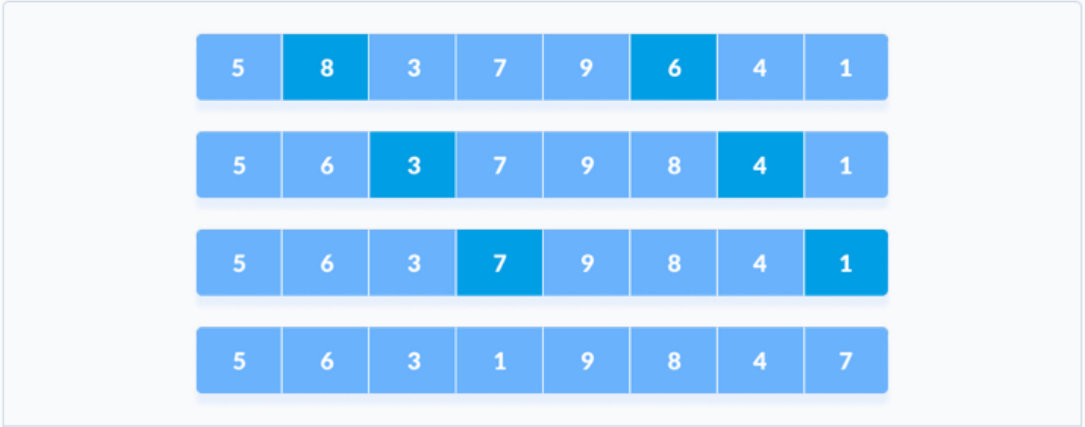
   a. The 0th element is compared with the `4th` element.

   b. If the 0th element is greater than the `4th` one then, the `4th` element is first stored in `temp` variable and the `0th` element (ie. greater element) is stored in the `4th` position and the element stored in `temp` is stored in the `0th` position.



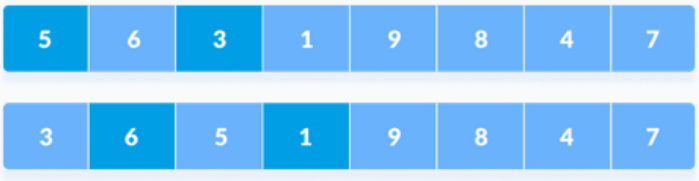
Rearrange the elements at n/2 interval

This process goes on for all the remaining elements.



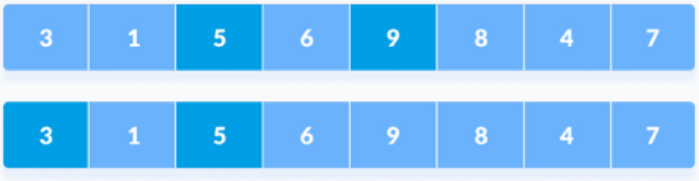
Rearrange all the elements at n/2 interval

3. In the second loop, an interval of `N/4 = 8/4 = 2` is taken and again the elements lying at these intervals are sorted.



Rearrange the elements at n/4 interval

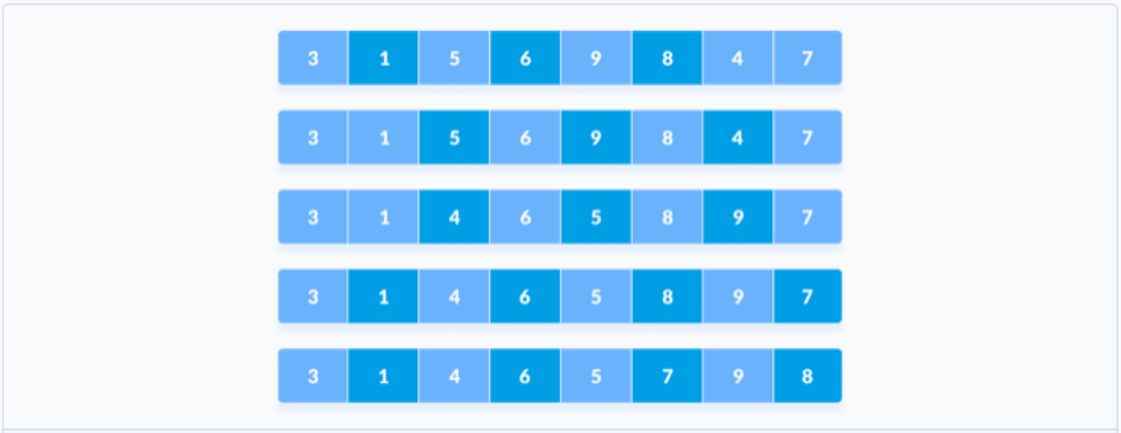
You might get confused at this point.



All the elements in the array lying at the current interval are compared.

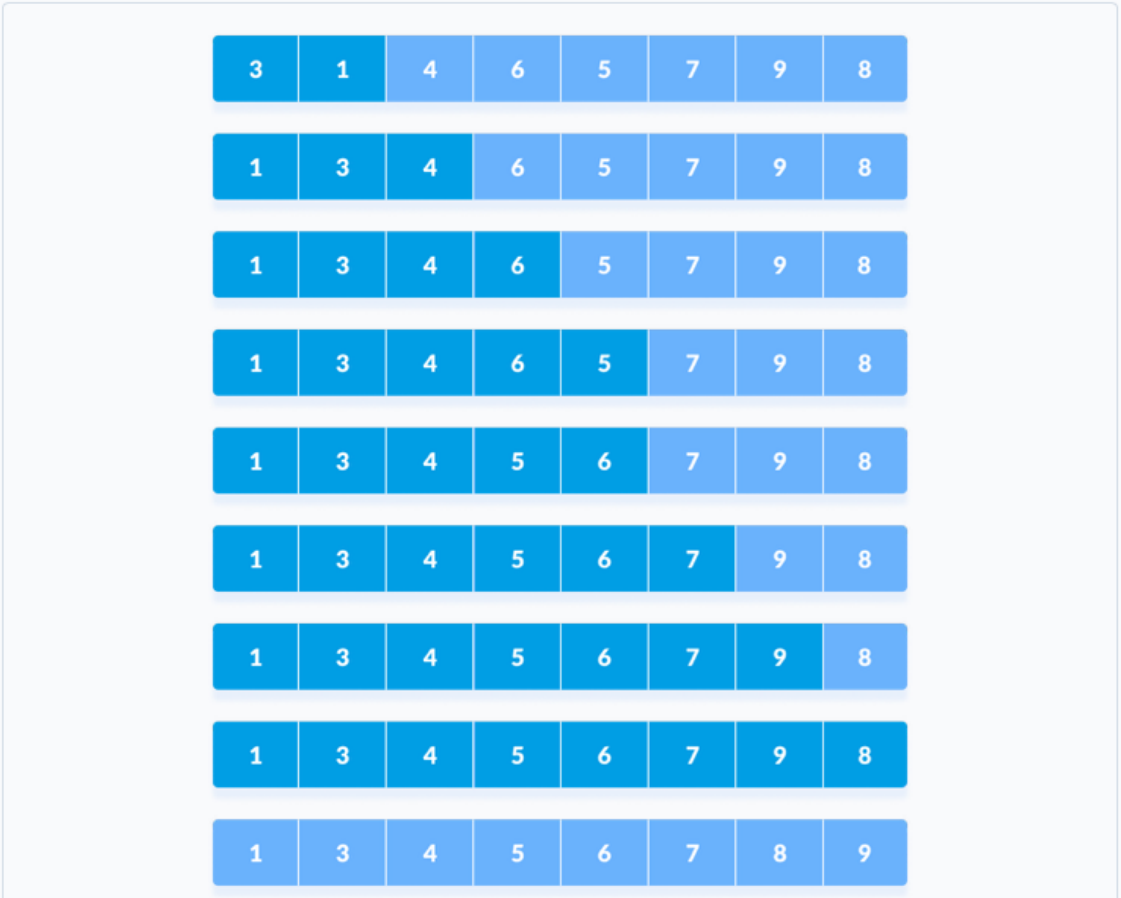
The elements at `4th` and `2nd` position are compared. The elements at `2nd` and `0th` position are also compared. All the elements in the array lying at the current interval are compared.

4. The same process goes on for remaining elements.



Rearrange all the elements at n/4 interval

5. Finally, when the interval is `N/8 = 8/8 =1` then the array elements lying at the interval of 1 are sorted. The array is now completely sorted.



Rearrange the elements at n/8 interval

Pseudocode

```
shellSort(array, size)
  for interval i <- size/2n down to 1
    for each interval "i" in array
        sort all the elements at interval "i"
end shellSort
```

Analysis

- **Worst Case Complexity**: less than or equal to $O(n^2)$
  Worst case complexity for shell sort is always less than or equal to $O(n^2)$.

  According to Poonen Theorem, worst case complexity for shell sort is $\theta(N\log N)^2/(\log\log N)^2)$ or $\theta(N\log N)^2/\log\log N)$ or $\theta(N(\log N)^2)$ or something in between.
- **Best Case Complexity**: $O(n*\log n)$
  When the array is already sorted, the total number of comparisons for each interval (or increment) is equal to the size of the array.
- **Average Case Complexity**: $O(n*\log n)$
  It is around $O(n^{1.25})$.

The complexity depends on the interval chosen. The above complexities differ for different increment sequences chosen. Best increment sequence is unknown.

Shell Sort Applications

1. Replacement for insertion sort, where it takes a long time to complete a given task.

2. To call stack overhead we use shell sort.

3. when recursion exceeds a particular limit we use shell sort.

4. For medium to large-sized datasets.

5. In insertion sort to reduce the number of operations.

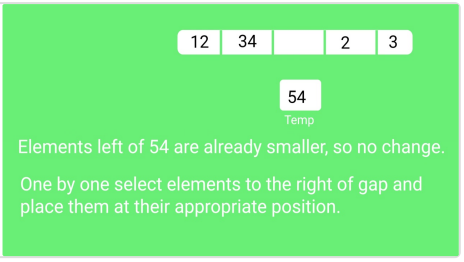
📌 Java implementation can be found under Implementation_Java folder

📚 References

Shell Sort - GeeksforGeeks

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles, quizzes and practice/competitive

https://www.geeksforgeeks.org/shell-sort/



Shell Sort Algorithm

Shell Sort Algorithm - Shell sort is a highly efficient sorting algorithm and is based on insertion sort algorithm. This algorithm avoids large shifts as in case of insertion sort, if

https://www.tutorialspoint.com/data_structures_algorithms/shell_sort_algorithm.htm

✍🏻 Author → Serhat Kumas

https://www.linkedin.com/in/serhatkumas/

SerhatKumas - Overview

Computer engineering student who loves coding in different fields instead of focusing on a one spesific area. - SerhatKumas

https://github.com/SerhatKumas