Shell sort

B) a variation of insertion Sort, just changed to allow for the exchange of far items. You would make a array h-sorted for a large value of h and would keep reducing the value of h until it is 1.

C) in this sorting algorithm we sort by comparing values a distance apart rather than the adjacent value and example of this would be this

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 35 | 56 | 3 | 4 |

Here is our array and we would start with a gap of n/2 (which is 2) and n is the size of the array. We will then select the elements right of the gap and place it in its correct positions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 35 | temp | 3 | 4 |

We take the data in the index position of 2 and compare it to the values to the left and as it is > than the values to the left it is already in the correct position

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 35 | 56 | temp | 4 |

Next we compare 3 with arr[3-2] = 35 as 34 > 2 we shift it to arr[gap+1 = 3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 |  | 56 | 35 | 4 |

Temp = 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 3 | 56 | 35 | 4 |

We then continue this process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 35 | 56 | 3 | temp |

Compare 4 with arr[4-2] = 56 and since 56 > 4 shift it to arr[gap+1=4]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 3 | 4 | 35 | 56 |

Start again from a gap of n/2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13 | 3 | temp | 35 | 56 |

Compare 2 with arr[2-2]= 13 since 4 < 13 we shift it to arr[gap+1 = 2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 3 | 13 | 35 | 56 |

Now that the entire right side is sorted a new gap is calculated or it is now 1(n/4) which in this case starts us at arr[1] and we continue the process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | temp | 13 | 35 | 56 |

Since 3 < 4 we move it

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 4 | 13 | 35 | 56 |

The gap is now reduced to 0 and the array has been sorted

D) it uses arrays

E) code example:

class ShellSort

{

static void printArray(int arr[])

{

int n = arr.length;

for (int i=0; i<n; ++i)

System.out.print(arr[i] + " ");

System.out.println();

}

int sort(int arr[])

{

int n = arr.length;

for (int gap = n/2; gap > 0; gap /= 2)

{

for (int i = gap; i < n; i += 1)

{

int temp = arr[i];

int j;

for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)

arr[j] = arr[j - gap];

arr[j] = temp;

}

}

return 0;

}

public static void main(String args[])

{

int arr[] = {13, 35, 56, 3, 4};

System.out.println("Array before sorting");

printArray(arr);

ShellSort ob = new ShellSort();

ob.sort(arr);

System.out.println("Array after sorting");

printArray(arr);

}

}