# CST 370
# Programming Assignment (Sorting Algorithm 2)

In this assignment you will implement a sorting algorithm, which has not been covered in class. We will call this sorting algorithm CoolSort(). We will take advantage of insertion sort for designing this sorting algorithm. The description of the sorting algorithm is as follows.

We will first choose a decreasing sequence of numbers that ends at 1. For example, let us consider the sequence of step sizes H = 5, 3, 1. You can choose any decreasing sequence. Note that the first element of the sequence is less than the number of elements in the array.

For each H, sort sub-arrays that start at arbitrary element and include every $H^{th}$ element using insertion sort. For example, consider the following array

a = [ 62 83 18 53 07 17 95 86 47 69 25 28].

An example run of CoolSort with gaps 5, 3 and 1 is shown below.

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | 62 | 83 | 18 | 53 | 07 | 17 | 95 | 86 | 47 | 69 | 25 | 28 |
| H = 5 | 17 | 28 | 18 | 47 | 07 | 25 | 83 | 86 | 53 | 69 | 62 | 95 |
| H = 3 | 17 | 07 | 18 | 47 | 28 | 25 | 69 | 62 | 53 | 83 | 86 | 95 |
| H = 1 | 07 | 17 | 18 | 25 | 28 | 47 | 53 | 62 | 69 | 83 | 86 | 95 |

The first pass, 5-sorting, performs insertion sort on separate subarrays ($a_1$, $a_6$, $a_{11}$), ($a_2$, $a_7$, $a_{12}$), ($a_3$, $a_8$), ($a_4$, $a_9$), ($a_5$, $a_{10}$). For instance, it changes the subarray ($a_1$, $a_6$, $a_{11}$) from (62, 17, 25) to (17, 25, 62). The next pass, 3-sorting, performs insertion sort on the subarrays ($a_1$, $a_4$, $a_7$, $a_{10}$), ($a_2$, $a_5$, $a_8$, $a_{11}$), ($a_3$, $a_6$, $a_9$, $a_{12}$). The last pass, 1-sorting, is an ordinary insertion sort of the entire array ($a_1$,..., $a_{12}$).

As the example illustrates, the subarrays that CoolSort operates on are initially short; later they are longer but almost ordered.

**Though unintuitive, it can be shown that the above algorithm has a runtime of $O(N^{3/2})$ in comparison to selection sort which has a runtime of $O(N^2)$. That is why this algorithm is cool!**

**Sample Test Case 1**

**Input** = [2, 5, 6, 4, 10, 9, 8, 1, 10, 5] and **H** = [5, 3,1]

**Output** = [1, 2, 4, 5, 5, 6, 8, 9, 10, 10]

**Sample Test Case 2**

**Input** = [2, 5, 9, 4, 10, 7, 8, 1, 11, 5] and **H** = [5, 2,1]

**Output** = [1, 2, 4, 5, 5, 7, 8, 9, 10, 11]

## Grading

I will download your code on my computer and execute it. If your code does not compile, you may lose more than 50% of your points (based on my discretion). If your code compiles, but still produces incorrect results you may still lose more than 30% of your points (based on my discretion).

Your code should have the following characteristics for you to get full points on the assignment

1. Compile without error.
2. Produce correct output.
3. Good programming structure.
4. Comments. (Title, Abstract, Author, ID, and Date are mandatory.)
5. Meaningful and related variable names.

## Extra credit

You will receive extra credit equal to 10% of your score if you submit a video (a link) explaining how you implement it (as well as some running samples of your program). Note that there is a separate place for you to submit the video link and that is where extra credit will be recorded. It should be submitted before the submission is closed (i.e., two days after the submission due date), but not subject to the late submission penalty. I will initiate a new topic on Discussion Forum to publish the video link to the class.

## What to turn in?

Submit your source programs **and 'HomeworkSubmission_yourlastname.pdf'** as a single zipped file ***'Program5_yourfullname'***on iLearn. For the program, please include only the source files needed to compile and run successfully.

**If you do not submit the above mentioned documents in the format specified your assignment will not be graded.**

**Homework Submission_yourlastname.pdf**
For each homework problem, you are expected to submit screenshots of the results obtained from running your code. You should also explain what each screenshot means and why the result on the screenshot is correct.

This link explains how to take screenshots in Mac and Windows.
http://www.take-a-screenshot.org/