Homework 2

Out: 9.16.21
Due: 9.27.21

1. [Asymptotic comparison, 25 points]
For each of these problems enter “yes” or “no” indicating whether A is O, o, Θ, ω, Ω of B. Justify your answers.

A	B	O	o	Θ	ω	Ω
4n log n	n log n + 5					
$2^{\log n}$	$(\log n)^2$					
$4{,}000{,}000^{2000}$	$\mathrm{Log}_{58}(n)$					
100,000	$\sum_{i=0}^{\infty} \frac{30}{5^i}$					
$\sum_{i=0}^{n} \frac{200}{n}$	$\sqrt{n}$					

2. [Asymptotics, 25 points]
Place the following functions from asymptotically smallest to largest. When two functions have the same asymptotic order, put an equal sign between them. Provide an explanation for your ordering.

$$1, n^3, n^{n^n}, 0, \frac{n}{4}, n^9 + n + 2, \sqrt[3]{n}, (n+1)^n, \sum_{k=1}^{\lg n} \frac{n}{3^k}, \left(1 + \frac{1}{n}\right)^n, \prod_{k=1}^{n} \left(1 - \frac{1}{k^3}\right), \ln n$$

3. [Algorithmic intuition, 50 points]
Write and briefly explain the following C++ function:

*void MaxDecSeq (int *nums, int len);*

that accepts an integer array, *nums*, containing *len* > 0 positive integers, and prints out, on a single line, the maximal length decreasing sequence in the input array. A decreasing sequence is defined as a sequence of numbers decreasing in magnitude (so should not include duplicates). These do not need to be located in adjacent cells of the input array, but do need to be located in increasing indices of the array. If there exists more than one maximal length decreasing sequence in the array, either sequence is a valid solution, but only print out one sequene.

For example, if [*nums*] contains [1 5 7 3 2 4] and *len*==6, the function should print the sequence 5 3 2 (or 7 3 2).
If [*nums*] contains [2 23 3 10 1 4 7] and *len*==7, the function should print the

sequence 23 10 1 (or 23 10 7).

Submit your solution, in a single file, *Problem3.cpp*, containing your function only. Your code should compile and run with the provided *Problem3.h* and *main.cpp* files on the lab computers. Try to make your function as efficient as you can.