Data Structures
BITL1

Worksheet 2

Fall 2020

10 September 2020

---

1. **Warm up:** Give descriptions for the following sets without the dots "...".

   (a) $\{0, 1, 2, 3, \dots\}$

   (b) $\{2, 4, 6, \dots\}$

   (c) $\{1, 3, 5, \dots\}$

   (d) $\{-10, -5, 0, 5, 10, 15, \dots\}$

   (e) $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \dots\}$

   (f) $\{0, 1, -1, 2, -2, \dots\}$

   (g) $[0, 1] \cup [2, 3] \cup [4, 5] \cup \cdots$

   (h) $[0, 1] \cap [0, 1/2] \cap [0, 1/3] \cap \cdots$

2. Consider the pseudocode below, which takes as input a set of numbers $X = \{x_1, \dots, x_n\}$.

```
1  for i = n, n − 1, ..., 2:
2    for j = 1, 2, ..., i − 1:
3      x = x_j
4      if x > x_{j+1}:
5        x_j = x_{j+1}
6        x_{j+1} = x
```

   (a) How many times is line 3 called?

   (b) What is an upper bound on the number of times line 5 is called?

   (c) In the boxes below, starting with X as given in step 0, write what X looks like every time the order of its elements changes.

step 0:

1	3	4	2

step 1:

step 2:

step 3:

step 4:

   (d) What do you think the code does to X?

This question is a review of "Big O" notation. Let $f, g\colon X \to \mathbf{R}$ be functions, for $X \subseteq \mathbf{R}$ and $a \in \mathbf{R}$. Then we say "*$f(x)$ is Big-O of $g(x)$ as x goes to a*", and write:

$$\text{“}f(x) = O(g(x)) \text{ as } x \to a\text{”, or “}f(x) = O(g(x)\text{”}$$

if a is clear from context, if there exists $\epsilon > 0$ and $M > 0$ such that $|f(x)| \leqslant M|g(x)|$ for all $x \in (a - \epsilon, a + \epsilon)$. If $a = \infty$, then the condition on x is changed to "for all $x > \epsilon$".

3. Let $n, m \in \mathbf{Z}_{>0}$.

   (a) Suppose that $f(x) = O(x^n)$ as $x \to 0$, and $g(x) = O(x^m)$ as $x \to 0$. Show that $f(x) + g(x) = O(x^k)$ as $x \to 0$, where $k = \min\{m, n\}$.

   (b) Suppose that $f(x) = O(x^n)$ as $x \to \infty$, and $g(x) = O(x^m)$ as $x \to \infty$. Show that $f(x) + g(x) = O(x^\ell)$ as $x \to \infty$, where $\ell = \max\{m, n\}$.

   (c) Let $f\colon \mathbf{Z}_{>0} \to \mathbf{Z}_{>0}$ be the function that, for an input n, returns the number of times line 3 from question 2 is called, given the input set $\{1, 2, \dots, n\}$. Find the smallest function $g(n)$, such that $f(n) = O(g(n))$.

   Note 1: Even though "n" is used as an argument here, instead of "x" as in the definition of Big-O, you may assume that the two can be interchanged.

   (d) Do you think there exists an alogrithm that sorts a list of length n, that has running time $O(1)$? Why or why not?