CSC 411 Quiz 1 — Solutions

1. If f and g are functions defined on the integers, what does it mean that a function $f = O(g)$?

   Solution. The statement $f(n) = O(g(n))$ means that f is eventually bounded above by a constant multiple of g. Formally, there exist constants $c > 0$ and n_0 such that for all integers $n \geq n_0$,

   $$0 \leq f(n) \leq c\, g(n).$$

   Intuitively: for sufficiently large inputs, f does not grow faster than g up to constant factors.

2. What is the worst-case time complexity of the following function. Explain your choice of input size parameter and "basic operation":

```
def find(int x, List xs):
  xs = mergesort(xs)
  for (i = 0; i < length(xs); i += 1):
    if xs[i] == x:
      return 8 // yes I know
  return -1
```

   Solution. Let $n = \text{length}(xs)$ be the input size parameter.

   Choose the basic operation as a key comparison (either comparisons performed by mergesort, or the test $xs[i] == x$ in the loop). Then:

   - Sorting with mergesort costs $\Theta(n \log n)$ comparisons in the worst case.
   - The post-sort linear scan costs up to n equality checks in the worst case (if x is absent, or at the last position), so $\Theta(n)$.

   Total worst-case running time:

   $$T(n) = \Theta(n \log n) + \Theta(n) = \Theta(n \log n).$$

   Therefore, the worst-case time complexity is $\Theta(n \log n)$ (or $O(n \log n)$ upper bound).

3. **(extra credit)** Let ϵ be a *very* small positive constant. Order these functions by increasing asymptotic complexity. Indicate if there are any "ties":

   $$n^2, \quad 2^{2+n}, \quad n^{1+\epsilon}, \quad n \log(n), \quad 2^n.$$

   Solution. First simplify $2^{2+n} = 4 \cdot 2^n$. Multiplying by a constant does not change asymptotic class, so

   $$2^{2+n} = \Theta(2^n).$$

   Next compare the polynomial-type terms:

- $n \log n$ grows faster than linear, but slower than $n^{1+\epsilon}$ (for any fixed $\epsilon > 0$ and sufficiently large n).
- Since ϵ is very small and positive, $1 + \epsilon < 2$, so $n^{1+\epsilon}$ grows more slowly than n^2.

Finally, exponentials eventually dominate polynomials, so 2^n grows faster than n^2.

Hence increasing order is:

$$n \log n \ < \ n^{1+\epsilon} \ < \ n^2 \ < \ 2^n \sim 2^{2+n}.$$

The only tie is 2^n with 2^{2+n} (same asymptotic class).