COEN 179 - Homework #1

Tamir Enkhjargal

April 2020

Question #1.

For the rest of the quarter, do you prefer synchronous lectures or asynchronous lectures?

A good mix of both, preferably a synchronous lecture to make sure the class is caught up to the same point, but other lectures can be online. A weekly Monday/Wednesday/Friday synchronous lecture is good.

Question #2.

Count the worst-case number of vector element comparisons ($V[mi] < V[i]$) made by `selection_sort` on vectors of size `n`:

```
void selection_sort(vector<string> &v) {
    for(int s = V.size(); s > 1; --s) {
        int mi(0);
        for(int i = 1; i < s; ++i)
            if(V[mi] < V[i])
                mi = i;
        swap(V[mi], V[s-1]);
    }
}
```

In the worst case, we have the that the outer for loop goes n times, and the inner loop goes s times each iteration. Therefore the if statement is hit: $(n) + (n-1) + (n-2) + ... + 3 + 2 + 1$ many times. We can use this math:
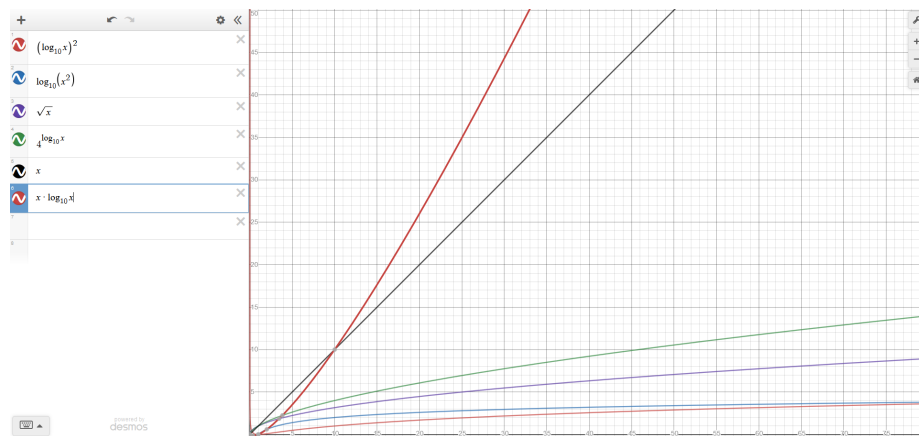
$$[(n) + (n-1) + (n-2) + ... + 3 + 2 + 1]$$
$$+[1 + 2 + 3 + ... + (n-2) + (n-1) + (n)]$$
$$=(n+1) + (n+1) + (n+1) + ... + (n+1)$$
$$= \frac{n+n^2}{2}$$

Therefore the if statement has $\frac{n+n^2}{2}$ comparisons. This means this is O(n^2). The `swap` only gets hit $n-1$ times (O(n)) in worst-case scenario.

Question #3.

Rank the functions $\sqrt{n}$, $(lgn)^2$, $lg(n^2)$, $nlgn$, n, $4^{lg(n)}$ in terms of growth rate, starting with the slowest. Show your work.

$lg^2(n)$, $lg(n^2)$, $\sqrt{n}$, $4^{lg(n)}$, n, $nlgn$

The easiest way to figure this out is to just graph the functions and see which ones approach higher or not. And continuing further in +x, $lg^2(n)$ plateaus off earlier than $lg(n^2)$. The other way is to check the derivative of each these functions at a certain x, and since the derivative is essentially the "growth rate", we can compare the numbers that way too.