Recitation Problems - Com S 311

Week of Jan 29^{th} - Feb 3^{rd}

1. **Prove $2^{n+1} \in O(2^n)$.**
   To show that $2^{n+1} \in O(2^n)$, we have to find positive constants c and n_0 such that $\forall n \geq n_0 \quad 2^{n+1} \leq c \cdot 2^n$. Since $2^{n+1} = 2 \cdot 2^n$, choose $c = 2$ and $n_0 = 1$. Then, $2^{n+1} \leq c \cdot 2^n$ for all $n \geq n_0$. Hence, $2^{n+1} \in O(2^n)$.

2. **Prove or disprove $3^n \in O(2^n)$.**
   We disprove $3^n \in O(2^n)$ by doing proof by contradiction.

   Let's assume $3^n \in O(2^n)$. This implies $3^n \leq c \cdot 2^n$ for some c, and n_0 s.t. $\forall n \geq n_0 \quad 3^n \leq c \cdot 2^n$
   $\Rightarrow (\frac{3}{2})^n \leq c$ for some c.

   As n increases, the left hand side increases but c remains the same. The LHS cannot be less than or equal to c for all valuations of n, leading to a contradiction. Hence, our assumption is false. Therefore, $3^n \notin O(2^n)$.

3. **Derive the worst-case runtime of the following loop structure as a function of n and determine its Big-O upper bound. You must show the derivation of the end result. Assume atomic operations take unit time.**

```
r = 0;

for(i = 1; i < n; i++)
{
    for(j = i + 1; j <= n; j++)
    {
        for(k = 1; k <= j; k++)
        {
            r = r + 1;  // Atomic operation taking constant time
        } // end k

    } // end j

} // end i
```

$$
\begin{aligned}
Runtime &= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{j} 1 \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} j \\
&= \sum_{i=1}^{n-1}(\sum_{j=1}^{n} j - \sum_{j=1}^{i} j) \qquad \left[using\ \ the\ \ formula \sum_{j=i+1}^{n} j = \sum_{j=1}^{n} j - \sum_{j=1}^{i} j\right] \\
&= \sum_{i=1}^{n-1}(\frac{1}{2}n(n+1) - \frac{1}{2}i(i+1)) \qquad \left[using\ \ the\ formula \sum_{x=1}^{n} x = \frac{1}{2}n(n+1)\right] \\
&= \frac{1}{2}n(n+1)(n-1) - \frac{1}{2}\sum_{i=1}^{n-1}(i^2+i) \\
&= \frac{1}{2}n(n+1)(n-1) - \frac{1}{2}(\sum_{i=1}^{n-1} i) - \frac{1}{2}(\sum_{i=1}^{n-1} i^2) \\
&= \frac{1}{2}n(n+1)(n-1) - \frac{1}{2}\cdot\frac{1}{2}n(n-1) - \frac{1}{2}(\sum_{i=1}^{n-1} i^2) \qquad \left[\sum_{x=1}^{n-1} x = \frac{1}{2}(n-1)((n-1)+1) = \frac{1}{2}n(n-1)\right] \\
&= \frac{1}{2}n(n-1)((n+1) - \frac{1}{2}) - \frac{1}{2}(\sum_{i=1}^{n-1} i^2) \\
&= \frac{1}{2}n(n-1)(n+\frac{1}{2}) - \frac{1}{2}(\sum_{i=1}^{n-1} i^2) \\
&= \frac{1}{2}n(n-1)(n+\frac{1}{2}) - \frac{1}{2}\cdot\frac{1}{6}n(n-1)(2n-1) \qquad \left[using\ \ the\ \ formula \sum_{x=1}^{n} x^2 = \frac{1}{6}n(n+1)(2n+1)\right]
\end{aligned}
$$

Note: you will need to substitute n with $n-1$ inside the formula

$$
\begin{aligned}
&= \frac{1}{2}n(n-1)\left[n + \frac{1}{2} - \frac{1}{6}(2n-1)\right] \\
&= \frac{1}{2}n(n-1)(n + \frac{1}{2} - \frac{1}{3}n + \frac{1}{6}) \\
&= \frac{1}{2}n(n-1)(\frac{2}{3}n + \frac{2}{3}) \\
&= \frac{1}{2}\cdot\frac{2}{3}\cdot n(n-1)(n+1) \\
&= \frac{1}{3}n(n-1)(n+1) \quad \text{which is } \underline{O(n^3)}
\end{aligned}
$$

4. **Derive the runtime of the following loop structure as a function of n and determine its Big-O upper bound. You must show the derivation of the end result. Assume atomic operations take unit time.**

```
i = 1;

while (i <= n)
{
    for (j = 1; j <= i; j++)
    {
        //<some constant number of atomic elementary operations>

    } // end j

    i = 2 * i;

} // end while
```

For each iteration of the outer loop, the inner loop will do i iterations. So if we consider R iterations of the outer loop:

Outer Loop Iteration	i Value at Beginning of Iteration	Number of Times Inner Loop Iterates
1	1	1
2	2	2
3	2^2	2^2
...	...	...
R	2^{R-1}	2^{R-1}

the summation of runtime will be: $1 + 2 + 2^2 + ... + 2^{R-1} = 2^R - 1$. If the outer loop iterates R times, then $2^{R-1} \leq n \Rightarrow 2^R \leq 2n$ as per condition of the while loop. Hence, the summation is $\leq 2n - 1$, which means that the runtime is $O(n)$.