Homework 5 Solutions

1. Consider the following algorithm:

```
ALG
val <- 1                                  // line 1
for i <- 5 to 20                          // line 2
   val <- i * i + i + val                 // line 3
display val                               // line 4
END
```

   (a) Write a summation for the number of times line 3 executes.
   $\sum_{i=5}^{20} 1$

   (b) Solve the above summation.

$$\begin{aligned}\sum_{i=5}^{20} 1 &= c(n-m+1) \\ &= 1(20-5+1) \\ &= 16\end{aligned}$$

   (c) What is the Big-O of this algorithm?
   $O(1)$

   (d) Write a summation for computing the **value** of `val`. (Summations can be used for many things.)
   $1 + \sum_{i=5}^{20} i^2 + i$

   (e) Solve the above summation; that is, what will `val` be at line 4? Use summation formulas.

$$\begin{aligned}1 + \sum_{i=5}^{20} i^2 + i &= 1 + \left[\sum_{i=0}^{20} i^2 - \sum_{i=0}^{4} i^2\right] + \left[\sum_{i=0}^{20} i - \sum_{i=0}^{4} i\right] \\ &= 1 + \left[\frac{20(20+1)(41)}{6} - \frac{4(5)(9)}{6}\right] + \left[\frac{20(21)}{2} - \frac{4(5)}{2}\right] \\ &= 1 + [2870 - 30] + [210 - 10] \\ &= 3041\end{aligned}$$

2. Consider this next algorithm:

```
ALG
enter x                              // line 1
enter y                              // line 2
bob <- 1                             // line 3
while bob < 20                       // line 4
   for jim <- 3 to bob               // line 5
      x <- y * x                     // line 6
   bob <- bob * 2                    // line 7
END
```

   (a) Write a summation for counting the number of times line 6 executes.

   $$\sum_{b=1}^{5} \sum_{j=3}^{2^{b-1}} 1$$

   Huh?

   This is tricky. Look at line 7. The variable `bob` *gets multiplied by 2 every time through the loop. Thus, the outer loop only executes 5 times. But the inner loop requires the value of* `bob`*. If you trace the algorithm, you see that* `bob` *gets the values 1, 2, 4, 8, and 16. This is the same as* 2^{b-1}*, where* $b \in \{1, 2, 3, 4, 5\}$*.*

   (b) Solve the above summation. Use summation formulas.

$$\begin{aligned}
\sum_{b=1}^{5} \sum_{j=3}^{2^{b-1}} 1 &= \sum_{b=1}^{5} (2^{b-1} - 3 + 1) && (1)\\
&= \sum_{b=1}^{5} (2^{b-1} - 2) && (2)\\
&= \left[\sum_{b=1}^{5} 2^{b-1}\right] - \left[\sum_{b=1}^{5} 2\right] && (3)\\
&= \left[\sum_{b=0}^{4} 2^{b}\right] - [2(5 - 1 + 1)] && (4)\\
&= \left[2^{4+1} - 1\right] - [10] && (5)\\
&= 21 && (6)
\end{aligned}$$

   In step 3, there is no rule that matches 2^{b-1}*. But you can rewrite the summation equivalently as shown in step 4. This now matches rule (1).*

   (c) What is the Big-O of this algorithm?

   $O(1)$

3. Solve the following summations:

   (a) $\sum_{i=1}^{50} 6i = 7650$

   (b) $\sum_{i=0}^{10} (4^i + 4i) = 1,398,321$

   (c) $\sum_{i=0}^{15} 10i^2 = 12,400$

4. Give the Big-O for each of the following:

   (a) $T(n) = 5n + 3n + n = O(n)$

   (b) $T(n) = (4n + 3)(2n + 8) = O(n^2)$

   (c) $T(n) = 24352342n^2 = O(n^2)$