<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">Assignment #2

Due Date: Saturday, September 9 at 11:59pm

Submit: eLearning

Late Policy: -10 points per hour late

Instructions: This is an individual assignment. Answers should be your own work.

Chapter 2

6 pts
1. In the definition of Big-O, why is the "for N &gt;= n0" needed?

6 pts
2. If f1(N) = 2N and f2(N) = 3N, why are they both O(N), since 3N is larger than 2N for N&gt;=1?

6 pts
3. a) For f1(N) = 2N and f2(N) = 3N:
       calculate f1(5) and f2(5), then f1(10) and f2(10). When N was doubled in each case, what happened to the result? Explain why this happens.
    b) For f1(N) = 2N*N and f2(N) = 3N*N:
       calculate f1(5) and f2(5), then f1(10) and f2(10). When N was doubled in each case, what happened to the result? Explain why this happens.

6 pts
4. Since Big-O notation is a mathematical tool for functions like f(N) or g(N), how is it applicable to algorithm analysis?

6 pts
5. Which grows faster, 2^n or n! ? Explain why.

10 pts (2 each)
6. Give the Big-O notation for the following expressions:

    a. 4n^5 + 3n^2 - 2
    b. 5^n - n^2 + 19
    c. (3/5)*n
    d. 3n * log(n) + 11
    e. [n(n+1)/2 + n] / 2

Questions 7-12 are 10 points each.
Assume numItems has the role of N, which may vary from one run to the next.

7. What is the Big-O running time for this code? Explain your answer.

```
    for (int i=0; i&lt;numItems; i++)
       System.out.println(i+1);
```

8. What is the Big-O running time for this code? Explain your answer.

```
    for (int i=0; i&lt;numItems; i++)
       for (int j=0; j&lt;numItems; j++)
          System.out.println( (i+1) * (j+1) );
```

9. What is the Big-O running time for this code? Explain your answer.

```
    for (int i=0; i&lt;numItems+1; i++)
       for (int j=0; j&lt;2*numItems; j++)
          System.out.println( (i+1) * (j+1) );
```

10. What is the Big-O running time for this code? Explain your answer.

```
    if ( num &lt; numItems )
```

```
      for (int i=0; i&lt;numItems; i++)
      {
         System.out.println(i);
      }
   else
      System.out.println("too many");
```

11. What is the Big-O running time for this code? Explain your answer.

```
     int i = numItems;
     while (i &gt; 0)
        i = i / 2;     // integer division will eventually reach zero
```

12. What is the Big-O running time for this code? Explain your answer.
    (You do not need to work out a recurrence formula).

```
     public static int div(int numItems)
     {
        if (numItems == 0)
           return 0;
        else
           return numItems%2 + div(numItems/2);
     }
```

Submit these files:
    hw2.doc (.doc can be .txt, .jpg, etc.)

</pre></body></html>