These are some example loop analysis problems that we will use for the first couple of lectures. We will use these examples to introduce some analysis techniques, some summation formulas, and big-Oh notation.

The goal in each problem is to determine the number of times that line indicated by “\// MARKER” is executed by each loop as a function of $N$.

1. for (int i=1; i<=N; ++i) 
   for (int j=i; j>0; --j) 
   {\// MARKER}

2. for (int i=1; i<N; i=i*2) 
   {\// MARKER}

3. for (int i=0; i<N; ++i) 
   for (int j=i; j>0; j=j/2) 
   {\// MARKER}

4. for (int i=1; i<N; ++i) 
   for (int j=0; j<N; j+=i) 
   {\// MARKER}

5. int i=0, sum=0; 
   while (sum < N) 
   { 
     sum = sum + i++; 
     \// MARKER 
   }

6. for (int i=1; i<N; i=i*2) 
   for (int j=0; j<i; ++j) 
   {\// MARKER}

For each loop, we want to estimate $T(N)$, its running time as a function of the unknown integer parameter $N$. Ideally we would like exact answers. However exact answers may be hard to find and more precise than we need, therefore we may trade-off some accuracy for an easier analysis.

This tradeoff is one motivation for big-Oh analysis, which we may use to express an upper bound on $T(N)$. For example, saying “$T(N) = O(N)$” says that for large enough values of $N$, $T(N)$ is upper bounded by some constant times $N$.

The book introduces $O(\cdot)$ notation in Section 2.4. We may also introduce $\Omega(\cdot)$ and $\Theta(\cdot)$ notation (which express lower bounds and/or upper bounds).

We will solve each of the above examples in class, I’ll post the solutions later once we’ve done them all.