# Problem Set 1

This problem set is due **in recitation** on **Friday, September 12.**

*Reading:* Chapters 1–4 (excluding §4.4); Kingston chapter.

There are **three** problems. Each problem is to be done on a separate sheet (or sheets) of three-hole punched paper. Mark the top of each sheet with your name, the course number, the problem number, your recitation section (i.e. letter), the date, and the names of any students with whom you collaborated.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode. English explanations should be used liberally in your pseudocode. For convenience, pseudocode may use standard arithemetic and logical operations, as well as loops and data structures like arrays. Pseudocode should be understandable to anyone who can program–it should **not** be ready to compile!
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.

---

**Problem 1-1. Recurrence Relations**

Solve the following recurrences. Give a $\Theta$ bound for each problem. If you are unable to find a $\Theta$ bound, provide as tight upper ($O$ or $o$) and lower ($\Omega$ or $\omega$) bounds as you can find. Justify your answers. You may assume that $T(1) = 1$.

**(a)** $T(n) = 7T(\frac{n}{2}) + n\sqrt{n}$

**(b)** $T(n) = 4T(\frac{n}{2}) + n^3 \log n$

**(c)** $T(2^n) = T(2^{n-1}) + 2^n$

**(d)** $T(n) = T(\frac{n}{9}) + T(\frac{n}{16}) + T(\frac{n}{25}) + \sqrt{n}$

**(e)** $T(n) = T(n-1) + n^3$

**(f)** $T(n) = T(\log n) + O(1)$

**(g)** $T(n) = 9T(\frac{n}{27}) + (\sqrt[3]{n})^2$

**Problem 1-2. Asymptotic Notation**

Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{16}$ of the functions satisfying $g_1 = \Omega(g_2)$, $g_2 = \Omega(g_3)$, $\ldots$, $g_{19} = \Omega(g_{16})$. Partition your list into equivalence classes such that $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$. (The function $\log^* n$ is discussed on pages 55-56 of CLRS.)

| | | | |
|---|---|---|---|
| $n^{2+\sin^2 n}$ | $\sum_{k=1}^{n} \frac{1}{k}$ | $n$ | $n^2$ |
| $n!$ | $3^{10000}$ | $3^n$ | $\log n$ |
| $2^n$ | $n^{\log \log n}$ | $n^3$ | $(\log n)^{\log n}$ |
| $n \log n$ | $(n+1)!$ | $\sum_{k=1}^{n} k$ | $\log(n!)$ |

**Problem 1-3. Sieve of Eratosthenes**

The Sieve of Eratosthenes, invented circa 200 B.C., is an algorithm to find all prime numbers between 2 and an input number $N$. The algorithm works as follows: we begin with a list of all integers from 2 to $N$. For each $m \leq N$, we cross out (i.e. mark as composite) each multiple of $m$ ($n \cdot m$ for $n \geq 2$) that is less than or equal to $N$. When this process terminates, only the prime numbers between 2 and $N$ are unmarked.

Below, we give pseudocode for this algorithm. At the beginning of the algorithm, every entry in the array $P$ is initialized to *true*, i.e. $P[i]$ is *true* for all $i$, $2 \leq i \leq N$. At the end of the algorithm, $P[i]$ is *true* iff $i$ is prime.

ERATOSTHENES-SIEVE$(N)$:
1 Let $P[i] \leftarrow true$ for all $i$ from 2 to $N$.
2 **for** $m \leftarrow 2$ **to** $N$:
3     $n \leftarrow 2$.
4     **while** $n \cdot m \leq N$:
5         $P[n \cdot m] \leftarrow false$.
6         $n \leftarrow n + 1$.
7     **endwhile**
8 **endfor**

The table below shows the values of the array elements $P[1 \ldots N]$ at the end of each **while** loop (line 7) during an execution of the algorithm run on input $N = 13$.

| **i** | $P[2]$ | $P[3]$ | $P[4]$ | $P[5]$ | $P[6]$ | $P[7]$ | $P[8]$ | $P[9]$ | $P[10]$ | $P[11]$ | $P[12]$ | $P[13]$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | T | T | T | T | T | T | T | T | T | T | T |
| 2 | T | T | **F** | T | **F** | T | **F** | T | **F** | T | **F** | T |
| 3 | T | T | F | T | **F** | T | F | **F** | F | T | **F** | T |
| 4 | T | T | F | T | F | T | **F** | F | F | T | **F** | T |
| ... | T | T | F | T | F | T | F | F | F | T | F | T |
| 13 | T | T | F | T | F | T | F | F | F | T | F | T |

Prove that the ERATOSTHENES-SIEVE algorithm is correct; that is, prove that upon termination, $P[i]$ is *true* iff $i$ is prime.

**Hint**: You can use the following pre-condition and post-condition and you can prove the suggested loop invariant for the **for** loop (line 2). Let $S_i$ represent the statement: $P[i]$ is *true* iff $i$ is prime.

**Pre-Condition:** $N \geq 2$.

**Post-Condition:** $S_i, \forall i$ such that $2 \leq i \leq N$.

**Loop Invariant:** $S_i, \forall i$ such that $2 \leq i \leq m$.