1. Homework
Due 1/22/09 before class

1. Code snippets (4 points)
For each of the two code snippets below give their Θ-runtime depending on \( n \).
Justify your answers.

(a) (3 points)
\[
\text{for} (i=n; \ i>=1; \ i=i-3) \{
\text{for} (j=n; \ j>=1; \ j=j/3) \{
\text{for} (k=3*n; \ k>=1; \ k--) \{
\text{print(" ");}
\}
\}
\}
\]

(b) (1 point)
\[
\text{for} (i=2; \ i<=n; \ i=i*i) \{
\text{print(" ");}
\}
\]

2. \( O \) and \( \Omega \) (4 points)
Prove the following, using the definitions of \( O \) and \( \Omega \):

- (2 points) \( 5n^3 + 3n + 2 \in O(n^3) \)
- (2 points) \( 5n^3 + 3n + 2 \not\in \Omega(n^4) \)

3. Selection sort (7 points)
Consider sorting \( n \) numbers stored in array \( A \) by first finding the smallest element of \( A \) and exchanging it with the element in \( A[1] \). Then find the second smallest element of \( A \), and exchange it with \( A[2] \). Continue in this manner for the first \( n-1 \) elements of \( A \). This algorithm is known as selection sort.

- (2 points) Write pseudocode for this algorithm.
- (2 points) What loop invariant does this algorithm maintain? Argue (informally) why this loop invariant will help prove the correctness of the algorithm.
- (1 point) Why does the algorithm need to run for only the first \( n-1 \) elements, rather than for all \( n \) elements?
- (2 points) Give best-case and worst-case running times (and example inputs attaining these runtimes) of selection sort in Θ-notation.

Flip over to back page  \[\rightarrow\]
4. **Big-Oh ranking (14 points)**

Rank the following functions by order of growth, i.e., find an arrangement $f_1, f_2, \ldots$ of the functions satisfying $f_1 \in O(f_2)$, $f_2 \in O(f_3)$, \ldots. Partition your list into equivalence classes such that $f$ and $g$ are in the same class if and only if $f \in \Theta(g)$. For every two functions $f_i, f_j$ that are adjacent in your ordering, prove shortly why $f_i \in O(f_j)$ holds. And if $f$ and $g$ are in the same class, prove that $f \in \Theta(g)$.

\[3n^3 + 4n^4, \quad n \log^2 n, \quad n^3, \quad \log \log n, \quad 2^n, \quad \log^2 n, \quad \sqrt{n}, \quad \sqrt{n}, \quad \log n, \quad n, \quad n \log n, \quad 2^n+2, \quad 4^n, \quad \log \sqrt{n}\]

As a reminder: $\log^2 n = (\log n)^2$ and $\log \log n = \log(\log n)$. Bear in mind that in some cases it might be useful to show $f(n) \in o(g(n))$, since $o(g(n)) \subset O(g(n))$. If you try to show that $f(n) \in o(g(n))$, then it might be useful to apply the rule of l’Hôpital which states that

\[
\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{f'(n)}{g'(n)}
\]

if the limits exist; where $f'(n)$ and $g'(n)$ are the derivatives of $f$ and $g$, respectively.