Qualifying Examination Spring 2008
Algorithms

Do all problems. No books or notes may be used. Your grade depends on the correctness, completeness and style of your answer.

Problem 1

a) (5 points): Define precisely what $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, and $f(n) = \Theta(g(n))$ mean.

b) (15 points): Prove that $\log_2 n! = \Theta(n \log n)$.

Problem 2

a) (15 points): Suppose that a_1, a_2, $\ldots$, a_n, and s are positive integers. Find the most efficient algorithm you can to determine whether there are $i, j, i \neq j$ such that $a_i + a_j = s$. The more efficient the algorithm, the more credit you get.

b) (25 points): Suppose that b_1, b_2, $\ldots$, b_n, and s are positive integers. Find the most efficient algorithm you can to determine whether there are i, j, $i \leq j$, such that $\sum_{k=i}^{j} b_k = s$. The more efficient the algorithm, the more credit you get.

Problem 3 The 3-Dimensional Matching (**3DM**) problem is defined as follows:

> **Instance**: A set $M \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets having the same number q of elements.
>
> **Question**: Does M contain a *matching*, that is, a subset $M' \subseteq M$ such that $|M'| = q$ and no two elements of M' agree in any coordinate?

and the Subset Sum (**SS**) problem is defined as follows:

> **Instance**: $n + 1$ positive integers, c_1, c_2, $\ldots$, c_n, and s.
>
> **Question**: Do there exist $q \leq n$ indices $1 \leq i_1 < i_2 < \ldots < i_q \leq n$ such that $\sum_{k=1}^{q} c_{i_k} = s$?

a) (5 points): Define the complexity class NP, and define what it means for a problem to be NP-hard.

b) (10 points): Show that **3DM** and **SS** are both in NP.

c) (25 points): Using the fact that **3DM** is NP-hard, show that **SS** is NP-hard.