Problem 1.
Under which circumstances is speed more important than efficiency when designing a parallel algorithm? (Give an example to highlight your answer).

Problem 2.
Consider two algorithms for solving a problem of size $N$, one that runs in $N$ steps on an $N$-processor machine and one that runs in $\sqrt{N}$ steps on an $N^2$ processor machine. Which algorithm is more efficient?

Problem 3.
Give an CREW algorithm for solving the problem of multiplying an $n \times n$ matrix $A$ and vector $x$ in $O(\lg n)$ time. How many processors does your algorithm require? How much work does it require? Comparing your algorithm to its sequential counterpart, what is its efficiency and speedup? (Use $O(\cdot)$ notation to describe various results).

Problem 4.
Make the algorithm in Problem 3 to work on an EREW PRAM. What is its running time?

Problem 5.
You are given $n$ binary values $X_1, \ldots, X_n$. Find the logical OR of these $n$ values in constant time on a CRCW PRAM with $n$ processors.

Problem 6 (extra credit).
For the algorithm in Problem 3, let us assume that we have only $p$ processors available where $p < n$ and $n$ is a multiple of $p$. How fast can we solve the matrix-vector multiplication problem on an EREW PRAM? Explain. Express parallel time, speedup and efficiency in terms of $n$ and $p$. 