1. Prove that if $(1/k)$th of the time spent executing an algorithm involves operations that must be performed sequentially, then an upper limit on the speedup achievable by executing the algorithm on parallel processors is $k$.

2. Devise a CREW (concurrent read, exclusive write) PRAM algorithm to multiply two $n \times n$ matrices, where $n = 2^k$. Suppose the running time of the optimal sequential algorithm for matrix multiplication is $O(n^{2.6})$, is your algorithm cost-optimal? Explain your answer.

3. Consider the following idea for a PRAM algorithm to merge two sorted lists, say $A$ and $B$, each of size $n/2$, using $n$ processors. Assume that all the $n$ elements are distinct. Associate each processor with one element of one of the lists, so that each of the $n$ elements is associated with distinct processors. Suppose a processor is associated with an element in list $A$. It knows the position of its data item in list $A$ (the index of the item in $A$), and hence knows the number of elements in list $A$ smaller than its data item. It then performs a binary search on the elements of list $B$ to figure out how many of the elements in $B$ are smaller than its data item. Thus it can figure out the position of its data item in the merged list.

Using the above idea, design a CREW PRAM algorithm for merging two sorted lists, each of size $n/2$. Derive the running time of your algorithm. Explain whether your algorithm is cost-optimal or not.

4. Using the above, design an $O(\log^2 n)$ PRAM algorithm to sort a list of $n$ distinct integers. Explain whether your algorithm is cost-optimal.