

6. We consider the problem of multiplying two  $n$  by  $n$  matrices.
1. Design a parallel algorithm that runs in time  $n$  on a CREW PRAM with  $n^2$  processors. What is the efficiency of this algorithm?
  2. Design a parallel algorithm that runs in time  $O(\log n)$  time on a CREW PRAM with  $n^3$  processors. What is the efficiency of this algorithm?
  3. Design a parallel algorithm that runs in time  $O(\log n)$  time on a CREW PRAM with  $n^3/\log n$  processors. What is the efficiency of this algorithm?
  4. Design a parallel algorithm that runs in time  $O(\log n)$  time on a EREW PRAM with  $n^3/\log n$  processors. What is the efficiency of this algorithm? HINT: Recall problem 3.

7. Design a parallel algorithm that given a polynomial  $p(x)$  of degree  $n$  and an integer  $k$  computes the value of  $p(k)$ . Your algorithm should run in time  $O(\log n)$  on a EREW PRAM with  $O(n/\log n)$  processors. Assume that the polynomial is represented by its coefficients.

8. We consider the problem of computing  $F_n$ , the  $n$ th Fibonacci number, given an integer  $n$  as input. Show how to solve this problem in time  $O(\log n)$  on a EREW PRAM with  $n$  processors. Make the unrealistic assumption that  $F_n$  will fit within one word of memory for all  $n$ , that is, assume that all arithmetic operations take constant time. Recall that  $F_n$  is defined by the following recurrence:  $F_0 = F_1 = 1$ , and  $F_n = F_{n-1} + F_{n-2}$  for  $n > 1$ . HINT: Note that for  $j > 0$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} F_j \\ F_{j-1} \end{bmatrix} = \begin{bmatrix} F_{j+1} \\ F_j \end{bmatrix}$$

Extra Credit: Reduction 24. The input to the triangle problem is a subset  $W$  of the Cartesian product  $X \times Y \times Z$  of sets  $X, Y$  and  $Z$ , each of cardinality  $n$ . The problem is to determine if there is a subset  $U$  of  $W$  such that 1) every element of  $X$  is in exactly one element of  $U$ , 2) every element of  $Y$  is in exactly one element of  $U$ , and 3) every element of  $Z$  is in exactly one element of  $U$ . Here's a story version of the same problem. You have disjoint collections of  $n$  pilots,  $n$  copilots, and  $n$  flight engineers. For each possible triple of pilot, copilot, and flight engineer, you know if these three people are compatible or not. Your goal is to determine if you can assign these  $3n$  people to  $n$  flights so that every flight has one pilot, one copilot, and one flight engineer that are compatible. Show that this problem is NP-hard using a reduction from 3SAT.