CS124 Section for PS10

abhi shelat

April 30, 1997

Our Solution

For part (a), we have an overkill solution in $O(n^3 \lg(n))$. For part (b), the final solution runs in $O(n^4)$, so it is not really a practical thing. Be careful with debugging.

Fast Exponentiation Trick

First, we talked about an algorithm in class that allows you to compute a^n rather quickly. Here is the specific algorithm:

FAST-EXPONENTIATION(D, n)
(1) $D' \leftarrow I$
(2) let $\langle n_k, n_{k-1}, \ldots, n_0 \rangle$ be the binary representation of n
(3) **for** $i = k$ **to** 0
(4) $D' \leftarrow D'^2$
(5) **if** $b_i = 1$ **then** $D' \leftarrow D'D$

The loop will run at most $\lg(n)$ times, and each iteration of the loop requires at most 2 matrix multiplications. Hence, the runtime of the algorithm, in terms of multiplications, is $O(2\lg(n))$.

Misc Details

1. If you are going to change your group, please let me know as soon as possible. Otherwise, I assume you are going to be with the same group as before.

2. We will have presentations on Wednesday, May 7, 1997. The order will be specified on the webpage. Please check it out this weekend, and let me know of any conflicts.

3. Please make your presentations concise, succint, and easy to follow.

4. Raquel will be teaching a section on May 7th that will cover Sorting networks.