Analysis and Design of Algorithms (B) '21
Midsem

March 3, 2021

1 An alternate fast multiplication

Recall the fast multiplication algorithm from class - given two n-digit numbers a, b, we can use divide and conquer and the identity

$$(x + y) \times (z + w) = xz + yw + xw + yz$$

for some $n/2$-digit numbers x, y, z, w, to find the product $a \times b$ faster than $O(n^2)$. This approach has the (very minor) disadvantage that the sum of two $n/2$-bit numbers (as in $x + y$ or $z + w$ in the above identity) can have $n/2 + 1$ bits which is slightly annoying. In this problem, your goal is to get around this issue by using instead the following identity (due to Knuth)

$$(x - y) \times (z - w) = xz + yw - xw - yz \tag{1}$$

Notice that now if x, y are $n/2$ bit numbers with $x > y$, then indeed $x - y$ has no more than $n/2$ bits.

1. Give pseudo-code for a divide and conquer algorithm for multiplying two n-digit numbers faster than $O(n^2)$ *using the identity* (1).
2. Write (with justification) a recurrence for the time complexity of the above algorithm.
3. Solve the recurrence and find the time complexity.

2 Find the tighest possible asymptotic behavior of $T(n)$ defined as

$$T(n) = T(\lceil n/3 \rceil) + T(\lceil 3n/5 \rceil) + 100n, T(1) = 1$$

3 Greedy Counter examples

The interval coloring problem, as done in the last class, is as follows - given a set of n intervals denoted by their start and end times (assume all start and end times are distinct), color the intervals using a minimum number of colors, so that any two intersecting intervals receive distinct colors.

Consider the following greedy algorithm. Sort the intervals in *decreasing order of start times*. Let the possible colors be indexed as $1, 2, 3, \ldots$. Now, color the intervals in the above sorted order. When coloring any interval I, assign it the color with the smallest possible number which is not present in any interval already colored and intersecting with I.

Give a counter-example to show that this algorithm may not be optimal.

4 No three in a row

You are given n balls arranged in a row. Each ball i has a value v_i. Give a polynomial time algorithm to pick a maximum value subset S of the balls so that *no three consecutive balls are in* S.

For example, say the ball values are

$$2 \quad 2 \quad 3 \quad 2 \quad 2$$

then, the maximum value subset has the first, second, fourth and the fifth ball, and their total value is $2 + 2 + 2 + 2 = 8$.

For this problem, you also need to write a proof of correctness of the recurrence.

5 Advertising budget optimization

Your company makes n items. If you invest (an integral) j amount of money to advertise item i, then you get a profit of $p_{i,j}$ from the sales of item i (you may assume that $p_{i,j}$ is non-decreasing in j for each i, though it doesn't really matter). You have a total advertising budget of B. Give an algorithm to allocate the budget to different items to maximize your total profit. Your algorithm should run in time polynomial in n, B (assume B is an integer).