

§1 Lecture 01-09

Trying to compute $x^n = e^{n \log(x)}$.
 $\exp(x, n)$.

$$\begin{cases} n = 0, & 1 \\ n = 1, & x \\ n > 1, n \% 2 = 0, & (\exp(x, n/2))^2 \\ n > 1, n \% 2 = 1, & (\exp(x, n/2))^2 * x \end{cases}$$

Let T_n be the time to compute x^n . Let $T_n = \underbrace{1}_{\text{Overhead}} + T_{n/2}$

Divide and conquer approach. Break down problem into a series of subproblems of smaller size, and then marry the solutions of the subproblems for the final output.

Example 1.1 (Chip Testing, Corman Exercise 4.5)

Problem. Factories cranking out billions of chips, far too many for humans to check by hand.

We are given a tester which takes two chips as an input. Each of these chips give an opinion of the other chip.

$$A \text{ --- Tester --- } B$$

Good chips do not lie, but bad chips cannot be trusted.

Options

GG: Both chips say good, so either they are both good, or they are both bad.

BB:

BG:

GB:

Model: Number of uses of a tester (Oracle)

Given: n chips. g represents the set of good chips and B represents the set of bad chips. $|g| + |B| = n$. It is known that $|g| > |B|$.

Wanted: Identify g using the tester $O(n)$ times.

Solution: If we have one good chip, is that very valuable? Absolutely because then you could pass the rest of the chips ($n - 1$ tests) through the machine, knowing that it doesn't lie. So the problem is really to find 1 good chip in $O(n)$.

Procedures

1. Take one chip and test it against all other chips. If the vote is greater than or equal to 50%, then it is good. This procedure takes $n - 1$ time. Repeating this procedure until success costs pessimistically $(n - 1) + (n - 2) + \dots + (n/2 - 1)$. A.K.A. worst case is $\Theta(n^2)$. $\geq n/2 \times n/2$ and $\leq n/2 \times n$.

2. If n is even, then we can make $n/2$ pairings. A percentage of these will be GG , a percentage will be BB , and a percentage will be BG/GB .

Take all BB , BG , and GB , and throw them out. Then take all the GG pairs, and throw out the right side.

At least of the two chips in each of the pairs that are thrown out must be bad. Therefore $\geq 50\%$ bad.

Therefore, in the GG section, the goods are bigger than the bads, especially in this section $|g| \geq |B| + 2$ (Pigeon Hole Principle). Having them, and we still have that

$$\frac{|g|}{2} \geq \frac{|B|}{2} + 1$$

T_n is the time to solve the problem

Worst case: $T_n \leq n/2 + T_{n/2}$, and $T_1 = 0$.

In case that n is odd, you have an extra $n - 1$ test. You do procedure 1

Example 1.2 (Multiplying two n-bit integers)

n bits. (if one of them has less than n bits you just add zeros)

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ \times \quad 1\ 0\ 1 \\ \hline 1\ 1\ 0\ 1 \\ 1\ 1\ 0\ 1\ 0 \\ \hline 1\ 1\ 1\ 1\ 0\ 1 \end{array}$$

Addition can be seen as a $2n \times n$ matrix. Time = $\Theta(n^2)$. n rows, $2n$ wide.

Kanatsuba method.

$$\begin{aligned} a &= \underbrace{a_1}_{n/2 \text{ bits}} \times 2^{n/2} + \underbrace{a_2}_{n/2 \text{ bits}} \\ b &= \underbrace{b_1}_{n/2 \text{ bits}} \times 2^{n/2} + \underbrace{b_2}_{n/2 \text{ bits}} \\ ab &= a_1b_1 \times 2^n + (a_1b_2 + a_2b_1)2^{n/2} + a_2b_2 \\ T_n &= \underbrace{4T_{n/2}}_{a_1b_1, a_1b_2, a_2b_1, a_2b_2} + \underbrace{\frac{3n}{2}}_{*} + \underbrace{12n}_{+} \end{aligned}$$

$$a_1b_2 + a_2b_1 = (a_1 - a_2)(b_2 - b_1) + a_2b_2 + a_1b_1$$

We already computed a_2b_2 and a_1b_1 . Now $T_n = 3T_{n/2} + an$. Constant in front of n will go up, but the $4 \rightarrow 3$ makes a huge huge difference. Now

$$T_n = \Theta(n^{\log_2 3})$$

Toom-cook method further improves on this. The game is on after this first break of $\Theta(n^2)$. You can get $T_n \leq 5T_{n/3} + cn$. It involves breaking into chunks of $n/3$ instead of $n/2$.

Best today: $O(n \log n)$. Published last year still being reviewed.

Example 1.3 (Mergesort)

Input is an array A , $A[1], \dots, A[n]$.

$$A' = A[1 - n/2]$$

$$A'' = A[n/2 + 1 - n]$$

$B' = \text{MERGESORT}(A')$ Cost is $T_{n/2}$ $B'' = \text{MERGESORT}(A'')$ Cost is $T_{n/2}$ *return*(merge of B' and B'')

$$T_n \leq T_{n/2} + T_{n/2} + n - 1 \approx 2T_{n/2} + n - 1$$

Note 1.4. Earlier we saw that $xT_{n/y}$ implies $T_n = \Theta(n^{\log_y 5})$ but this is not always true.

§1.1 Convex Hull

Can think of it as tightening a string around a set of points in \mathbb{R}^2 . From any point in convex hull you can drop line where are points are on one side.

Convincing you that finding convex hull is the same as merge sort.

First find left and right hull. Then find upper and lower hull.