Divide and conquer

Peter Hajnal

Bolyai Institute, University of Szeged, Hungary

2023 Fall

The basic idea of divide and conquer

- Your given a data. The problem is such that for any subset of the data it makes sense to propose the problem.
- The scheme is half the input (or divide into few groups) and handle the two halves separately.
- From the solutions for the parts we try to figure out the answer for the whole data. The mathematical essence is that very often combining the two (or more) partial solutions can be done efficiently.
- The best way to understand the scheme is to see examples.

Sorting

Sorting problem

Given n real numbers: $x_1, x_2, \ldots, x_n$. Find their ordered sequence.

We are lookinf for such a π permutation of the index set $[n] = \{1, 2, \ldots, n\}$ that $x_{\pi 1} \leq x_{\pi 2} \leq \ldots \leq x_{\pi n}$.

Our model is a comparison based computation. I.e. our only elementary step is comparison. When we analyze the algorithm we just count how many comparisons we make.

Divide

- In the case of $n = 2$, we need one comparison to sort.

- In the case of larger input size consider the number $x_1, x_2, \ldots, x_{\lceil n/2 \rceil}$ and the numbers $x_{\lceil n/2 \rceil + 1}, x_{\lceil n/2 \rceil} + 2, \ldots, x_n$ separately. Sort them using the idea recursively.

- Let $y_1 \leq y_2 \leq \ldots \leq y_{\lceil n/2 \rceil}$ be the sorted list of $x_1, x_2, \ldots, x_{\lceil n/2 \rceil}$. Let $z_1 \leq z_2 \leq \ldots \leq z_{\lfloor n/2 \rfloor}$ be the sorted list of $x_{\lceil n/2 \rceil + 1}, x_{\lceil n/2 \rceil} + 2, \ldots, x_n$.

- Combine the two sorted lists into one. Combining the sorted list of the y's and z's are called merging the two sorted lists. The heart of the divide and conquer sorting algorithm is the way of merging.

Merging

Theorem

Let $y_1 \leq y_2 \leq \ldots \leq y_k$ and $z_1 \leq z_2 \leq \ldots \leq z_\ell$ two ordered sequence. With at most

$$k + \ell - 1$$

comparisons we can merge them.

Proof: See recitation session.

The algorithm

Merge sort

(Case of small inputs) If $n \leq 2$, then sort them.

(Divide and use recursion) If $n \geq 3$, then Sort $x_1, x_2, \ldots, x_{\lceil n/2 \rceil}$ and $x_{\lceil n/2 \rceil + 1}, x_{\lceil n/2 \rceil} + 2, \ldots, x_n$ separately using Merge sort.

(Merge) Merge the two sequences using the algorithm from the recitation session.

The analysis of the algorithm

- Let $S(n)$ the minimal number of comparison that is enough to handle all inputs of size n.

- We know that $S(1) = 0$ and $S(2) = 1$.

The analysis of the algorithm (cont'd)

Theorem

$$S(n) \leq S\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + S\left(\left\lceil \frac{n}{2} \right\rceil\right) + n - 1.$$

The analysis of the algorithm: Conquer

Lemma

If $n = 2^k$

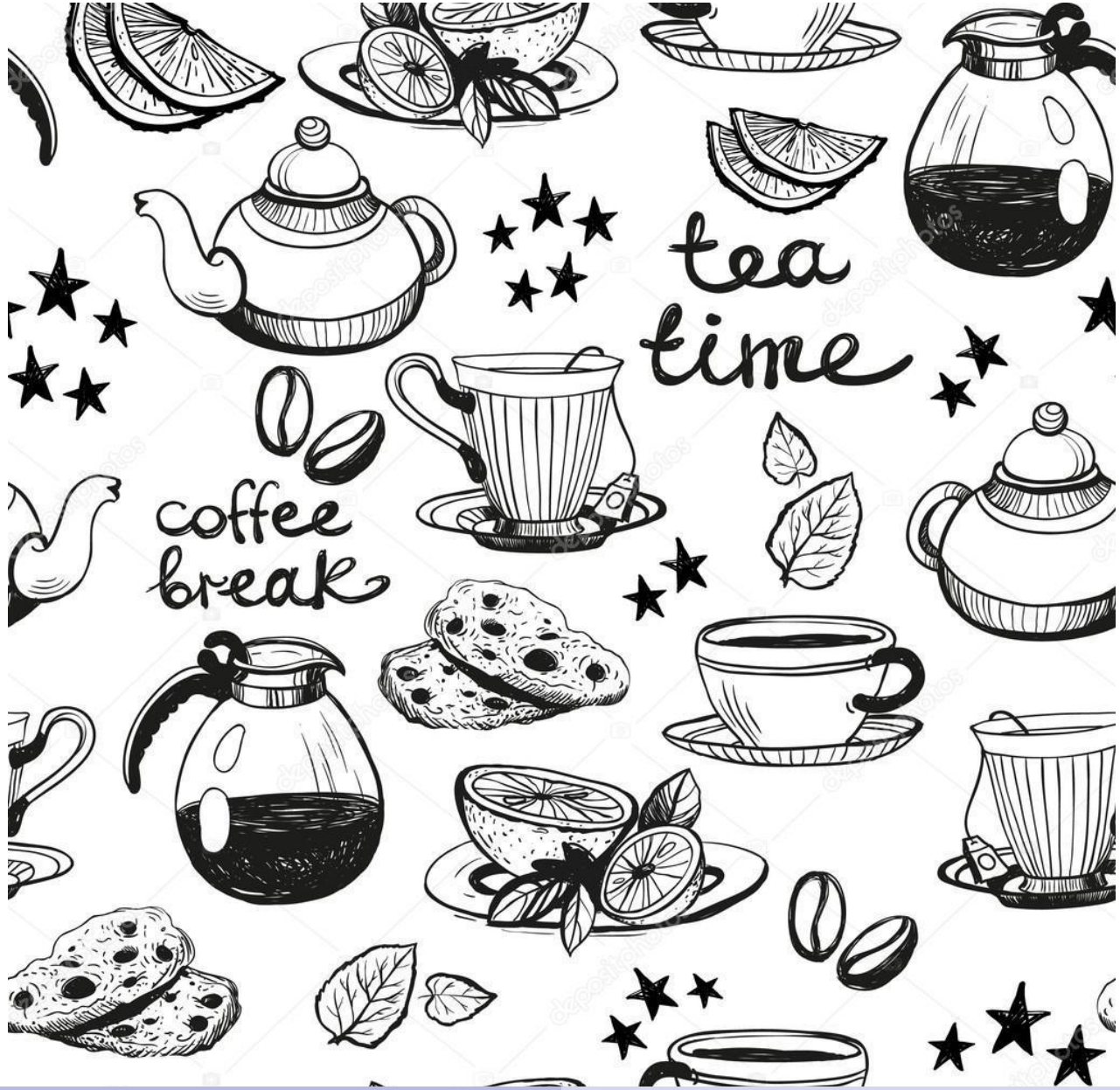
$$S(n) \leq (k-1)2^k + 1 = n(\log_2 n - 1) + 1.$$

Theorem

$$S(n) < 2n \log_2 n + 1 = \mathcal{O}(n \log n).$$

- This is the optimal order of magnitude (see recitation).

Break



The basic problem

The problem of multiplication

Given two n-digit numbers $x_1x_2\ldots x_n$ and $y_1y_2\ldots y_n$ (i.e. x_i's and y_j's are elements of $\{0,1,2,3,4,5,6,7,8,9\}$). Compute the digits of their product.

Why?

- Why we consider the multiplication?

- Summation, subtraction, dividing/multiplying by 2 are much easier. The well-known algorithm uses $\mathcal{O}(n)$ operations on digits.

- This is optimal.

The problem

• The well-known algorithm for multiplication takes $\mathcal{O}(n^2)$ operations on digits.

• In the XXth century, at the beginning of the 60's Kolmogorov conjectured that one can't do better. He proposed this as a problem to work on at his world-famous seminar.

• One of the participant, Karatsuba refuted the conjecture.

The naive "divide"

• The naive application of the Divide and conquer technique suggests the following. Think about the two input numbers x, and y as

$$x_{first} \cdot 10^{\nu} + x_{second}, \qquad \textit{and} \qquad y_{first} \cdot 10^{\nu} + y_{second}.$$

• The product of x and y

$$x_{first}y_{first} \cdot 10^{2\nu} + (x_{second}y_{first} + x_{first}y_{second}) \cdot 10^{\nu} + x_{second}y_{second}.$$

• We need four multiplications on half-size numbers, and additional simple operations (summations).

• Important to note that multiplication with 10^{ν} and $10^{2\nu}$ do not use any digit operations.

The analysis of the naive Divide and conquer

• Introduce the function $N(n)$: the maximal number of digit operations is needed to multiply two n-digit numbers. (The minimal guarantee for n-digits numbers.)

• We have

$$N(n) \leq 4 \cdot N\left(\frac{n}{2}\right) + \mathcal{O}(n).$$

• Using the $\mathcal{O}$ notation is laziness. The hidden constant can be determined, or can be easily bounded by 100.

Setback

• The "solution" of the inequality is $N(n) = \mathcal{O}(n^2)$. The bound matches the complexity the algorithm from elementary school.

The first idea of Karatsuba

- The computing the square of a given number has the same complexity as multiplication.
- Indeed

$$x \cdot y = \frac{(x+y)^2 - x^2 - y^2}{2},$$

hence multiplication of two numbers can be substituted by three computations of squares and "easy" side computations.

- Now on we consider the problem of squaring a number.

Divide and conquer for squaring

- The formula

$$(x_{first} \cdot 10^{\nu} + x_{second})^2 = x_{first}^2 \cdot 10^{2\nu} + x_{second}^2 + 2x_{first}x_{second} \cdot 10^{\nu}$$

is most promising. We need two squaring and one multiplication on half-size numbers.

The second idea

- The second idea is the same as the first:

$$(x_{first} \cdot 10^{\nu} + x_{second})^2 = x_{first}^2 \cdot 10^{2\nu} + x_{second}^2 + ((x_{first} + x_{second})^2 - x_{first}^2 - x_{second}^2) \cdot 10^{\nu}$$

The analysis

- Introduce the corresponding complexity function $M(n)$.
- The obvious bound is

$$M(n) \leq 3M\left(\frac{n}{2}\right) + \mathcal{O}(n).$$

- The solution is

$$M(2^k) = \mathcal{O}(3^k).$$

- The exact bound is mathematical induction.
- We obtain

$$M(n) = \mathcal{O}(n^{\log_2 3}).$$

- The algorithm for multiplication, thought on elementary school is not optimal.

Final remark

• Karatsuba's algorithm is far being optimal.

• There is "new" algorithm and its analysis gives that in $\mathcal{O}(n \log n)$ steps one can multiply two n-digit numbers.

• D. Harvey, J. van der Hoeven, Integer multiplication in time $\mathcal{O}(n \log n)$, Annals of Mathematics, 193(2), 2021, 563–617.

This is the end!

Thank you for your attention!