Recurrence equations

CS 146 - Spring 2017

Today

- Divide-and-conquer wrap-up
- Analysis of Karatsuba’s algorithm
- Analysis of divide-and-conquer algorithms
- Q&A

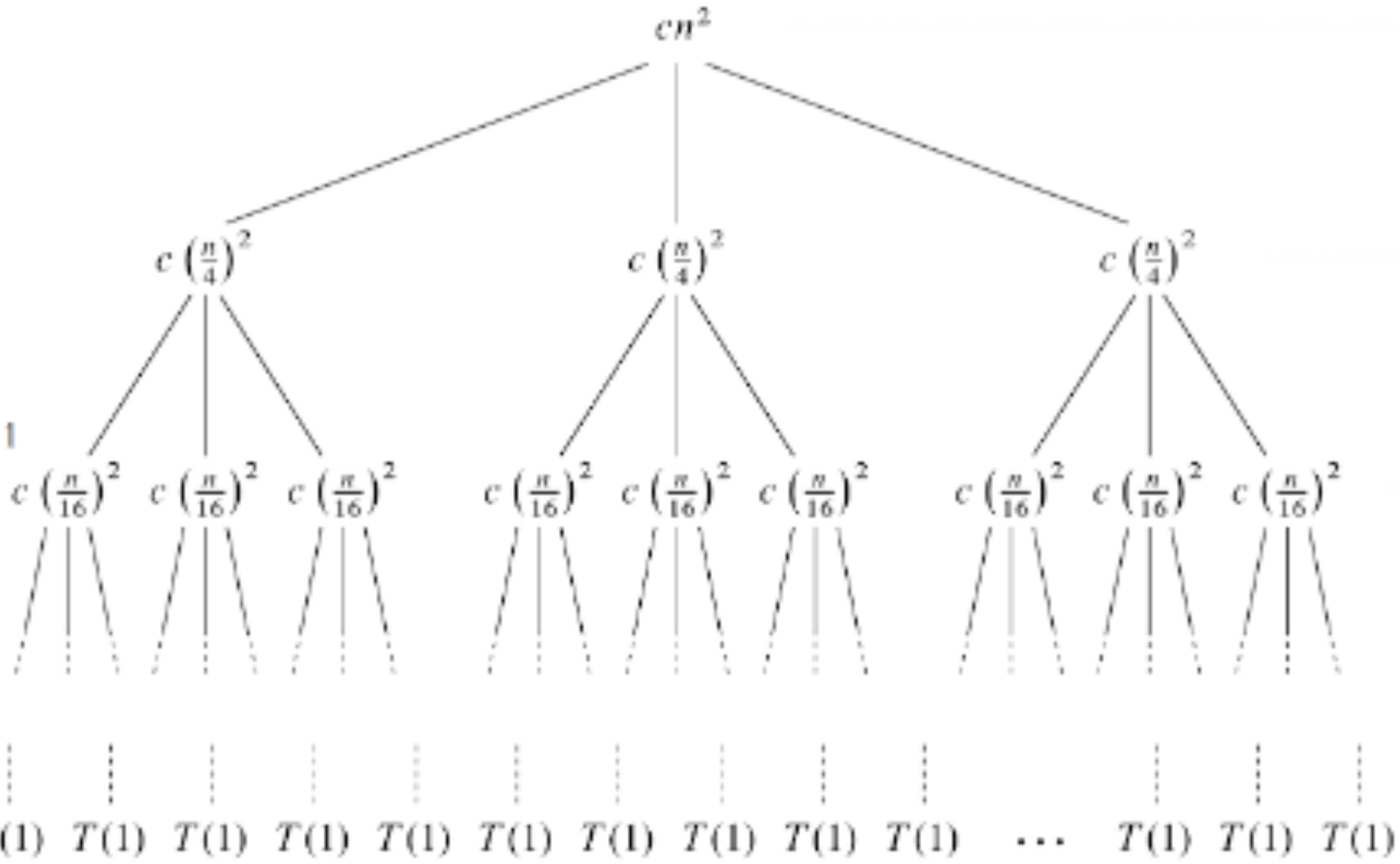
Recap: divide-and-conquer algorithms

- anatomy of a divide-and-conquer algorithm
  - divide: split problem into smaller subproblems
  - recurse: solve each subproblem recursively
  - conquer: combine the results of the subproblems
- D&C not inherently more efficient. Need
  - evenly split subproblems
  - efficient divide and conquer steps

Recap: divide-and-conquer algorithms

Problem	Non-D&C	D&C
sorting	insertion sort selection sort $O(n^2)$	mergesort quick sort O(n log n)
max-subarray-sum	brute force $O(n^3)$	O(n log n)
multiplication	grade school $O(n^2)$	Karatsuba $O(n^{\log_2 3})$

Divide-and-conquer algorithms in practice

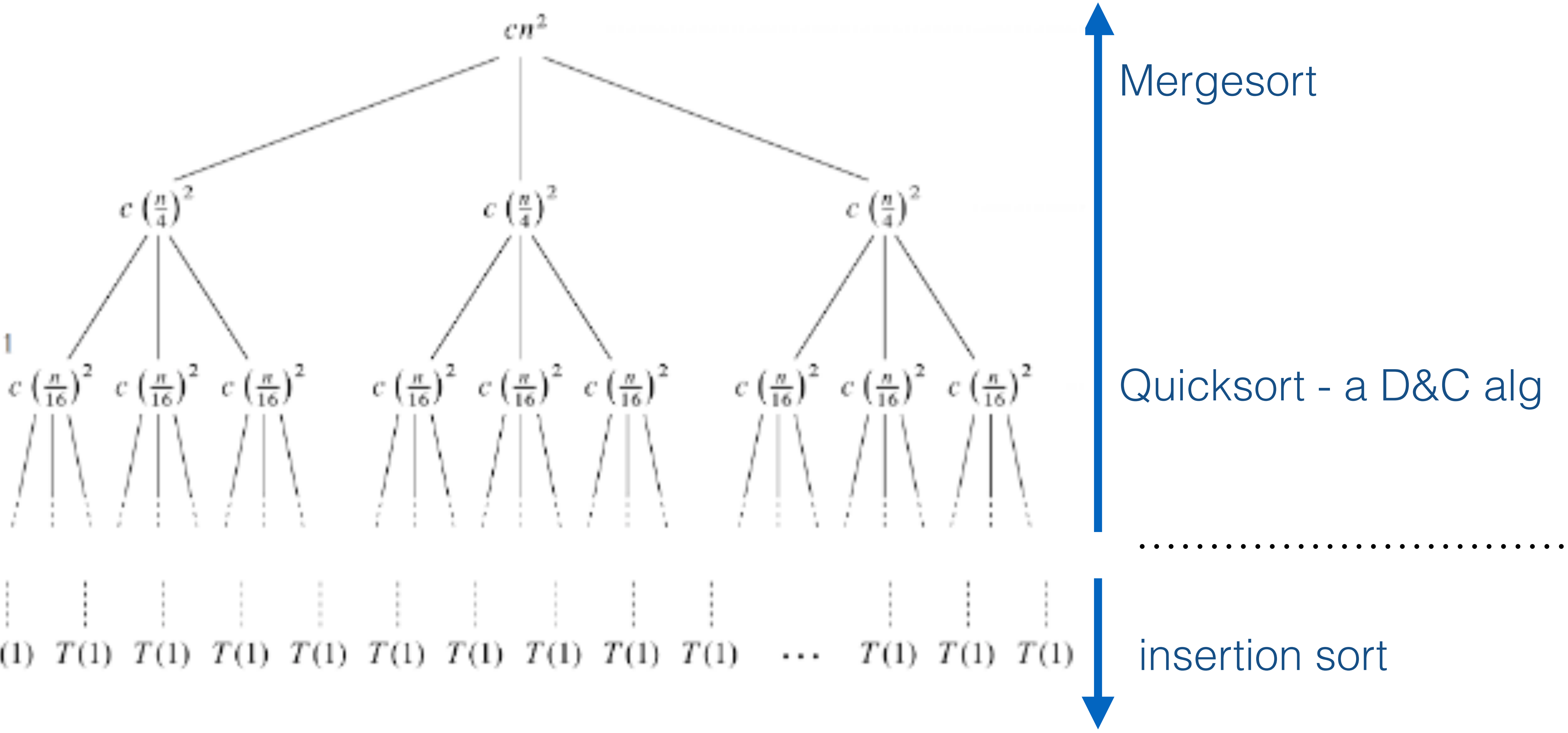

cn^2
$c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$
$T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ … $T(1)$ $T(1)$ $T(1)$

above a certain threshold size, asymptotic behavior dominate - use a D&C alg

below threshold size, constant overhead dominates - use simple arg

http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap08.htm

Java implementation of Arrays.sort


cn^2
$c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$ $c\left(\frac{n}{16}\right)^2$
(1) $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ $T(1)$ … $T(1)$ $T(1)$ $T(1)$

Mergesort

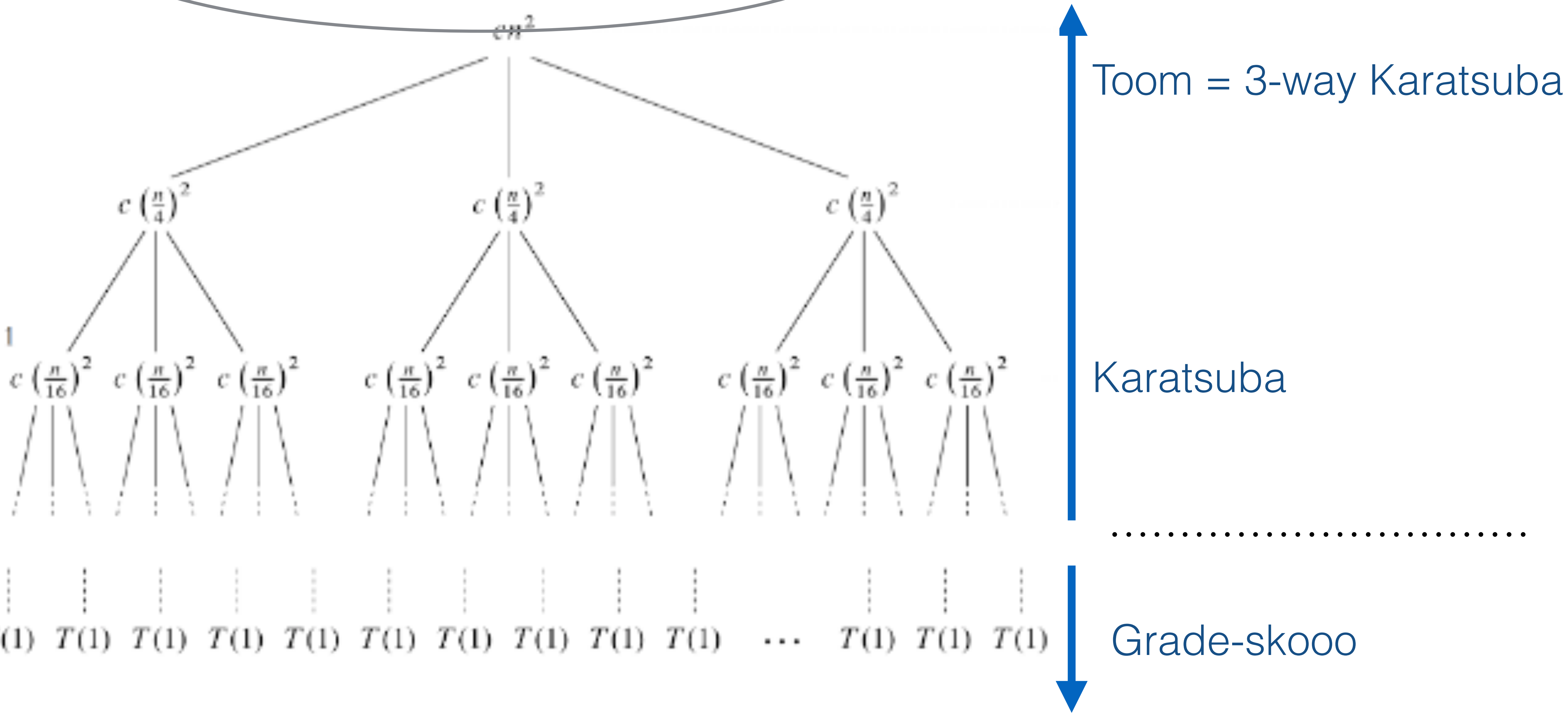
Quicksort - a D&C alg

insertion sort

http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap08.htm

GMP library multiplication

GNU multi-precision
(scientific computing)



Toom = 3-way Karatsuba

Karatsuba

Grade-skooo

http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap08.htm

- Why Karatsuba's running time is $O(n^{\log_2 3})$

What’s the time complexity?

```
void foo(int n) {
    if (n <= 1) return;
    for (int i = 0; i < n*n; i++)
        print “woof”;
    foo(n/3);
    foo(n/3);
    foo(n/3);
    foo(n/3);
}
```

Solving a recurrence

- total time =
  - sum from 1 to the number of tree levels of…
  - where at level i,
    - #nodes at this level, times,
    - the time taken to complete function call with given input size at this level

Is there any advantage to splitting 2-ways rather than 3-ways in mergesort?

Recurrence equation

- an equation where the unknown is a function T(n)
- as in algebra, there are multiple approaches to solve:
  - guess and check
  - **solve step by step** ← **tree method**
  - plug numbers into a formula ← master theorem

know this!