CS240E W24 Tutorial 0 additional note: Master theorem Jan. 15

Master theorem is a tool for asymptotic analysis of *divide-and-conquer* recurrences. A formal treatment of divide-and-conquer is **not** part of CS240E, but many algorithms we do see operate in its spirit.

A **divide-and-conquer** algorithm breaks down a problem into subproblems, recursively solves the subproblems, and then combines those solutions to a solution of the original problem. Divide-and-conquer solves many problems, e.g.

merge sort: to sort an array $A[1..n]$, sort $A[1..n/2]$ and $A[n/2..n]$ recursively, then merge the solutions;

Karatsuba's multiplication: to multiply two n digit numbers, multiply two $n/2$ digit numbers (three times), and add the products;

closest pair of points: given n points, find a pair of points with the smallest distance between them (details: CS341);

Fast Fourier Transform: multiply two polynomials of degree n (details: later in CS240E).

Let $\mathcal{A}$ divide-and-conquer algorithm, denote by $T(n)$ its worst-case runtime on an instance of size n. Suppose $\mathcal{A}$ divides the problem into three equally-sized subproblems, recursively solves the subproblems and then combines them in linear time. Then we may write

$$T(n) \leq 3T(\lceil n/3 \rceil) + cn$$

for some constant c and all sufficiently large n. Master theorem immediately resolves the simpler version of this recurrence:

$$T(n) \leq 3T(n/3) + n.$$

Note: results about simpler versions of recurrences do not automatically transfer to original version (in this case it would, but even then we need to adapt the proof). In particular, in A1Q2 we **must** be careful with simpler versions of recurrences.

Theorem (Master theorem). Suppose $T(n)$ satisfies

$$T(n) \le aT(n/b) + n^c$$

for all sufficiently large n. Then

if $\log_b a < c$, then $T(n) \in O\left(n^c\right)$,

if $\log_b a = c$, then $T(n) \in O\left(n^c \log n\right)$, and

if $\log_b a > c$, then $T(n) \in O\left(n^{\log_b a}\right)$.

Q1. Give a proof of Master theorem.

Q2. Use Master theorem to give an asymptotic upper bound for each of

1. $T(n) = 3T(n/2) + n^2$
2. $T(n) = 3T(n/4) + n \log n$
3. $T(n) = 5T(n/2) + n^2 \log^2 n$

Q3. Can we resolve

$$T(n) \le T(n-1) + cn$$

using Master theorem? Give an example of an algorithm whose worst-case runtime $T(n)$ satisfies this recurrence for all sufficiently large n.