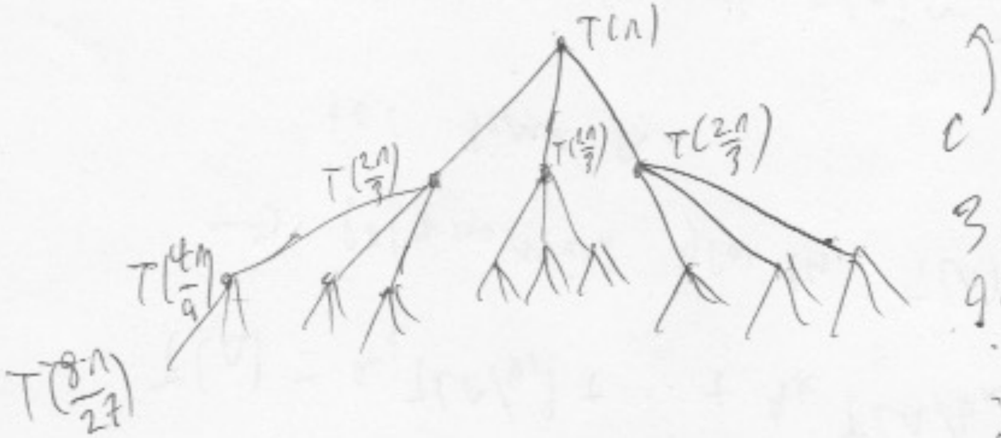
Stooge sort: Sorts an array by sorting the left $2/3$, then the right $2/3$, then the left $2/3$ again.

A: ~~Time~~

Running time $T(n)$ satisfies the recurrence

$$T(n) = 3T\left(\tfrac{2}{3}n\right) + \Theta(1).$$



$3^{k-1}c$

$T\left(\left(2/3\right)^{k} n\right)$

Height: $\log_{3/2}(n) + O(1) \rightarrow (2/3)^{k} \cdot n = 1$ (or 2).

Leaves: $\Theta\left(3^{\log_{3/2}(n)}\right) = \Theta\left(n^{\log_{3/2}(3)}\right)$

Nodes: $\Theta\left(n^{\log_{3/2}(3)}\right)$

$a^{\log_b(c)} = c^{(\log_c a)\cdot \log_b c} = c^{\frac{\log a}{\log c}\cdot\frac{\log c}{\log b}} = c^{\frac{\log a}{\log b}} = c^{\log_b a}$

E.g. $(\log n)^{\log n} = n^{\log\log n}$

Total Cost: $\Theta(n^{\log_{3/2}(3)})$.

Verify: Suppose $T(m) = \Theta(n^{\log_{3/2}(3)=\alpha})$ for $m<n$,

$c_1 m^\alpha \le T(m) \le c_2 m^\alpha - k$

Then $T(n) = 3 \cdot T(2/3\, n) + \cancel{O(1)}\, k$.

so $T(n) \le 3 \cdot [c_2(\frac{2n}{3})^\alpha - k] + k$

$\le 3 \cdot (2/3)^\alpha \cdot c_2 n^\alpha + k - 3k$

$= c_2 \cdot n^\alpha \cancel{+k} - 2k$

$\le c_2 n^\alpha - k$

and $T(n) \ge 3\, c_1 (\frac{2n}{3})^\alpha + k$

$\ge 3 \cdot (2/3)^\alpha \cdot c_1 n^\alpha$

$= c_1 n^\alpha$

$a=3$, $b=3/2$, $f(n) = \Theta(1)$. $\alpha = \log_{3/2}(3)$

$f(n) = O(n^{\alpha-\varepsilon})$ for $\varepsilon = \alpha > 0$, so

$T(n) = \Theta(n^\alpha)$.

$T(n) = 6 \cdot T(4n/9) + \Theta(n^2 \cdot (\log n)^0)$.

$a=6$, $b=\frac{9}{4}$, $f(n) = \Theta(n^2)$, $\alpha = \log_{9/4}(6)$.

$(9/4)^2 = \frac{81}{16} < 6 \Rightarrow \alpha > 2$, so we are in case 1.

Thus $T(n) = \Theta(n^\alpha)$.

Master method: Let $a \geq 1$, $b > 1$ be constants,

let $f(n)$ be an asymptotically positive function,

and let $T(n)$ satisfy the recurrence

$$T(n) = aT(n/b) + f(n),$$

where n/b is interpreted as $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Let $\alpha := \log_b a$.
Then $T(n)$ has the following asymptotic bounds:

(1) (Leaf-heavy) If $f(n) = O(n^{\alpha - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\alpha})$

(2) (Middle case) If $f(n) = \Theta(n^{\alpha} \log^{k} n)$ for some constant k, then

(a) If $k > -1$, then $T(n) = \Theta(n^{\alpha} \log^{k+1} n)$,

(b) If $k = -1$, then $T(n) = \Theta(n^{\alpha} \log \log n)$, and

(c) If $k < -1$, then $T(n) = \Theta(n^{\alpha})$

(3) (Root-heavy) If there is $c < 1$ such that for all sufficiently large n, $af(n/b) \leq c\, f(n)$, then $T(n) = \Theta(f(n))$.

Rough estimation:

$$T(n) = T(n-k_1) + T(n-k_2) + \cdots + T(n-k_r)$$

with at least two terms, then $T(n)$ grows exponentially.

$$T(n) = a_1 T(n/b_1) + \cdots + a_k T(n/b_k)$$

$\rightarrow$ polynomial growth, $T(n) = \Theta(n^\alpha)$ for some α.

$$T(n) = T(n-k) + f(n) \Rightarrow T(n) \approx \sum_{j>1}^{n} f(j)$$

- Solve different types of terms separately, then look at fastest growth.

$$T(n) = 2T(n-1) + 3T(7n/9) + 2^{\sqrt{n}}$$

$T(n) = 2T(n-1) \rightarrow T(n) = 2^n$

$T(n) = 3T(7n/9) \rightarrow T(n) \propto n^\alpha$

$T(n) = 2^{\sqrt{n}} \rightarrow T(n) = 2^{\sqrt{n}}$