CSE 332
Summer 2018
Section 7 – recurrence solutions

1. Show that quicksort with sequential partitioning, but parallel recursive sorting, has $\mathcal{O}(n)$, span by solving the recurrence relation shown in lecture:

$$T(n) = \begin{cases} T(n/2) + c_1 n & \text{if } n \geq 1 \\ c_2 & \text{othwerise} \end{cases}$$

Solution:

(a) Size of the input at level i: $\frac{n}{2^i}$.

(b) Number of nodes at level i: 1

(c) Work done at recursive level i: $1 \cdot c_1 \frac{n}{2^i}$

(d) Last level of the tree when $n/2^i = 1$ i.e. $\log_2(n)$.

(e) Base case work: $1 \cdot c_2$.

(f) Total work:

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\log(n)-1} c_1 \frac{n}{2^i} + c_2 \\
&= c_1 n \sum_{i=0}^{\log(n)-1} \frac{1}{2^i} + c_2 && \text{factoring} \\
&= c_1 n \frac{\left(\frac{1}{2}\right)^{\log n} - 1}{1/2 - 1} + c_2 && \text{finite geometric series formula} \\
&= c_1 n \frac{1/n - 1}{1/2 - 1} + c_2 && \text{logs and exponents are inverses} \\
&= c_1 n \left(\frac{1/n}{-1/2} - \frac{1}{-1/2} \right) + c_2 && \text{splitting fraction} \\
&= 2c_1 n - 2c_1 + c_2 && \text{simplification}
\end{aligned}$$

This is $O(n)$ as claimed.

2. Show that a completely parallel quicksort, (i.e. quicksort with parallel partition and recursion) has span $\mathcal{O}(log^2(n))$, by solving the recurrence relation shown in lecture:

$$T(n) = \begin{cases} T(n/2) + c_1 \log n & \text{if } n \geq 2 \\ c_2 & \text{otherwise} \end{cases}$$

.

Solution:

(a) Size of the input at level i: $\frac{n}{2^i}$.

(b) Number of nodes at level i: 1

(c) Work done at recursive level i: $1 \cdot c_1 \log\left(\frac{n}{2^i}\right)$

(d) Last level of the tree when $n/2^i = 1$ i.e. $\log_2(n)$.

(e) Base case work: $1 \cdot c_2$.

(f) Total work:

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\log(n)-1} c_1 \log\left(\frac{n}{2^i}\right) + c_2 \\
&= c_1 \sum_{i=0}^{\log(n)-1} \log\left(\frac{n}{2^i}\right) + c_2 && \text{factoring} \\
&= c_1 \sum_{i=0}^{\log(n)-1} \log(n) - \log(2^i) + c_2 && \log(ab) = \log a + \log b \\
&= c_1 \left[\sum_{i=0}^{\log(n)-1} \log(n) - \sum_{i=0}^{\log(n)-1} \log(2^i)\right] + c_2 && \text{split sums} \\
&= c_1 \left[\sum_{i=0}^{\log(n)-1} \log(n) - \sum_{i=0}^{\log(n)-1} i\right] + c_2 && \text{logs and exponents are inverses} \\
&= c_1 \left[\log^2(n) - \frac{\log(n)(\log(n)-1)}{2}\right] + c_2 && \text{closed forms} \\
&= c_1 \left[\frac{\log^2(n)}{2} + \frac{\log(n)}{2}\right] + c_2 && \text{algebra}
\end{aligned}$$

This is $O(\log^2(n))$ as claimed.

3. Show that completely parallel mergesort (i.e. mergesort with parallel merging and recursion) has span $\mathcal{O}(\log^3(n))$ by solving the recurrence shown in lecture:

$$T(n) = \begin{cases} T(n/2) + c_1 \log^2(n) & \text{if } n \geq 2 \\ c_2 & \text{otherwise} \end{cases}$$

Solution:

(a) Size of the input at level i: $\frac{n}{2^i}$.

(b) Number of nodes at level i: 1

(c) Work done at recursive level i: $1 \cdot c_1 \log^2\left(\frac{n}{2^i}\right)$

(d) Last level of the tree when $n/2^i = 1$ i.e. $\log_2(n)$.

(e) Base case work: $1 \cdot c_2$.

(f) Total work:

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\log(n)-1} c_1 \log^2\left(\frac{n}{2^i}\right) + c_2 \\
&= c_1 \sum_{i=0}^{\log(n)-1} \log^2\left(\frac{n}{2^i}\right) + c_2 && \text{factoring} \\
&= c_1 \sum_{i=0}^{\log(n)-1} \left[\log(n) + \log\left(2^{-i}\right)\right]\left[\log(n) + \log(2^{-i})\right] + c_2 && \text{because } \log(ab) = \log a + \log b \\
&= c_1 \sum_{i=0}^{\log(n)-1} \left[\log^2(n) - 2i\log(n) + i^2\right] + c_2 && \text{expanding multiplication} \\
&= c_1 \left[\sum_{i=0}^{\log(n)-1} \log^2(n) - \sum_{i=0}^{\log(n)-1} 2i\log(n) + \sum_{i=0}^{\log(n)-1} i^2\right] + c_2 && \text{splitting sum} \\
&= c_1 \left[\log^3(n) - 2\log(n)\frac{\log(n)(\log(n)-1)}{2} + \right. \\
&\qquad \left. \frac{(\log(n)-1)(\log(n))(2\log(n)-1)}{6}\right] + c_2 && \text{closed forms we'll give you.}
\end{aligned}$$

This is a valid closed form.

We'll simplify further to see the O() more easily

$$= c_1 \left[\log^2(n) + \frac{(\log(n)-1)(\log(n))(2\log(n)-1)}{6}\right] + c_2 \qquad \text{cancellation in first two terms}$$

This is $O(\log^3(n))$ as claimed.