Time complexity analysis of Largest sub-array sum

Dr. Anand M
Id: 102763
Assistant Professor
Department of Computer Science and Engineering
SRM Institute of Science and Technology

Max Subarray: Algorithm Efficiency

We finally have:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$

The recurrence has the same form as that for MERGESORT, and thus we should expect it to have the same solution T(n) = Θ(n lg n).

This algorithm is clearly substantially faster than any of the brute-force methods. It required some cleverness, and the programming is a little more complicated – but the payoff is large.

Time complexity analysis

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

$$\begin{aligned}
T(n) &\leq 2T(\frac{n}{2}) + cn \\
&\leq 2[2T(\frac{n}{4}) + c\frac{n}{2}] + cn = 4T(\frac{n}{4}) + 2cn \quad \text{1st expansion} \\
&\leq 4[2T(\frac{n}{8}) + c\frac{n}{4}] + 2cn = 8T(\frac{n}{8}) + 3cn \quad \text{2nd expansion} \\
&\vdots \\
&\leq 2^k T(\frac{n}{2^k}) + kcn \quad \text{kth expansion}
\end{aligned}$$

The expansion stops when $2^k = n$

$$\begin{aligned}
T(n) &\leq nT(1) + cn\log_2 n \\
&= O(n) + O(n \log n) \\
&= O(n \log n)
\end{aligned}$$

Time complexity analysis

- Theorem

$$T(n) = \begin{cases} O(1) & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + O(n) & \text{if } n \geq 2 \end{cases} \Rightarrow T(n) = O(n \log n)$$

- Proof
  - There exists positive constant a, b s.t. $T(n) \leq \begin{cases} a & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + b \cdot n & \text{if } n \geq 2 \end{cases}$
  - Use induction to prove $T(n) \leq 2b \cdot n \log_2 n + a \cdot n$
    - n = 1, trivial
    - n > 1, $\lceil \frac{n}{2} \rceil \leq \frac{n}{\sqrt{2}}$

$$\begin{aligned} T(n) &\leq T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + b \cdot n \\ &\leq 2b \cdot (\lceil n/2 \rceil \log_2 \lceil n/2 \rceil) + a \cdot \lceil n/2 \rceil + 2b \cdot (\lfloor n/2 \rfloor \log_2 \lfloor n/2 \rfloor) + a \cdot \lfloor n/2 \rfloor + b \cdot n \\ &\leq 2b \cdot (\lceil n/2 \rceil \log_2 \frac{n}{\sqrt{2}} \rceil) + a \cdot \lceil n/2 \rceil + 2b \cdot (\lfloor n/2 \rfloor \log_2 \frac{n}{\sqrt{2}}) + a \cdot \lfloor n/2 \rfloor + b \cdot n \\ &= 2b \cdot n(\log n - \log_2 \sqrt{2}) + a \cdot n + b \cdot n = 2b \cdot n \log_2 n + a \cdot n \end{aligned}$$

Inductive hypothesis