COMP 410-001; Fall 2009

$\mathrm{S}_k\mathrm{B}$

The runtime complexity $T(n)$ of **binary search** is characterized by the following recurrence:

$$\begin{aligned} T(1) &= c_1 \\ T(n) &= T(n/2) + c_2, \quad \text{for } n > 1 \end{aligned}$$

To solve this recurrence, let k denote $\lceil \log n \rceil$; equivalently, k is the smallest integer such that $2^k \geq n$.

Since $T(n)$ is a non-decreasing function of n, we have

$$\begin{aligned} T(n) &\leq T(2^k) \\ &= T(2^{k-1}) + c_2 \\ &= T(2^{k-2}) + c_2 + c_2 \\ &= T(2^{k-3}) + 3c_2 \\ &\cdots \quad \cdots\cdots \\ &= T(2^0) + kc_2 \\ &= c_1 + kc_2 \\ &= c_1 + \lceil \log n \rceil c_2 \\ &< c_1 + (\log n + 1)c_2 \\ &= c_1 + c_2 + c_2 \log n \\ &= O(\log n) \end{aligned}$$