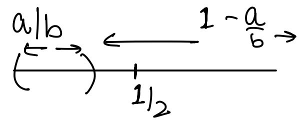
Q: Generalise that Binary search is better than any other division of $\left(\frac{a}{b}\right)$ where $a < b$

Soln:

Case 1: $\left(\frac{a}{b}\right) = \frac{1}{2}$. In that case it's just Binary Search

where Time: $O(\log_2 n)$.

Case 2: $\left(\frac{a}{b}\right) < \frac{1}{2}$. In that case



$1 - \frac{a}{b}$ forms the major chunk & the complexity is given as $T(n) = T\left\{\left(1 - \frac{a}{b}\right)n\right\} + k$ ← constant

i.e. $T(n) = T\left\{\left(\frac{b-a}{b}\right)n\right\} + k$

Now

$$n$$
$$\downarrow$$
$$n \cdot \left(\frac{b-a}{b}\right) = \frac{n}{(b/b-a)}$$
$$\downarrow$$
$$n \cdot \left(\frac{b-a}{b}\right)^2 = \frac{n}{(b/b-a)^2}$$
$$\downarrow$$
$$\vdots$$
$$\downarrow$$
$$1 = \frac{n}{(b/b-a)^{\log_{(b/b-a)} n}}$$

$\therefore$ No. of terms / height $= \log_{(b/b-a)} n + 1 \approx \log_{(b/b-a)} n$.

Now $\because$ a & b are non-negative numbers such that $b > a$ & $\left(\frac{a}{b}\right) < \frac{1}{2}$

$$\Rightarrow \frac{a}{b} - 1 < \left(-\frac{1}{2}\right)$$

$$\Rightarrow \frac{a-b}{b} < -1/2$$

$$\Rightarrow -\frac{(a-b)}{b} > 1/2$$

$$\Rightarrow \frac{b-a}{b} > 1/2$$

$$\Rightarrow \frac{b}{b-a} < 2.$$

$\therefore$ Complexity $= O\left(k \cdot \log_{(b/b-a)} n\right) = O\left(\log_{(b/b-a)} n\right)$

$\because \left(\frac{b}{b-a}\right) < 2 \Rightarrow \boxed{\log_{(b/b-a)} x > \log_2 x}$

Case 3: $\frac{a}{b} > 1/2$, then $T(n) = T\left(\frac{a}{b} \cdot n\right) + k$.

$$n \;—\; k$$
$$\downarrow$$
$$n \cdot \frac{1}{(b/a)} \;—\; k$$
$$\downarrow$$
$$n \cdot \frac{1}{(b/a)^2} \;—\; k$$
$$\downarrow$$
$$\vdots$$

Clearly $1 = \frac{n}{n} = n \cdot \frac{1}{(b/a)^{\log_{(b/a)} n}}$

$\therefore$ Height of this $= \log_{(b/a)} n + 1$

Now $\because$ $\frac{a}{b} > \frac{1}{2} \Rightarrow \frac{b}{a} < 2$

$\Rightarrow$ Time complexity in this case is

$$O(k \cdot \log_{b/a} n) = \boxed{O(\log_{b/a} n) > O(\log_2 n)}$$

Thus in all the case $O(\log_2 n)$ is the best we can get. Hence dividing into $1/2$ is the most suitable division.