Algorithm BinarySearch (L,x, first, last)
if first > last **then return** -1 $\}\ C_1$
else {
mid ← (first +last)/2
if x = L[mid] **then return** mid ⇐
else if x < L[mid] **then**
return BinarySearch (L,x,first,mid -1)
else return BinarySearch (L,x,mid +1,last
)

C_2

f(n) = number of operations performed by the algorithm in the worst case $(x \notin L)$ when the input has size n

$$f(0) = C_1$$

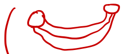
$$f(n) = C_2 + f\left(\frac{n-1}{2}\right)$$

Recurrence equation

Repeated Substitutions

$$f(0) = c_1$$

$$f(n) = c_2 + \underline{f\left(\frac{n-2^0}{2^1}\right)}, \quad n > 0$$

f() $= c_2 + f\Big($ ( $- 1) / 2 \Big)$

$$f\left(\frac{n-2^0}{2^1}\right) = c_2 + f\left(\frac{\frac{n-1}{2} - 1}{2}\right) = c_2 + f\left(\frac{n-2^0-2^1}{2^2}\right)$$

$$f\left(\frac{n-2^0-2^1}{2^2}\right) = c_2 + f\left(\frac{\frac{n-1-2}{2^2} - 1}{2}\right) = c_2 + f\left(\frac{n-2^0-2^1-2^2}{2^3}\right)$$

$$f\left(\frac{n-2^0-2^1-2^2}{2^3}\right) = c_2 + f\left(\frac{n-2^0-2^1-2^2-2^3}{2^4}\right)$$

$$f\left(\frac{n-2^0-2^1-2^2-2^3}{2^4}\right) = c_2 + f\left(\frac{n-2^0-2^1-2^2-2^3-2^4}{2^5}\right)$$

$$\vdots$$

$$f\left(\frac{n-2^0-2^1-\cdots-2^k}{2^{k+1}}\right) = c_2 + f\left(\underbrace{\frac{n-2^0-2^1-\cdots-2^{k+1}}{2^{k+2}}}_{=0}\right)$$

$$n-2^0-2^1-\cdots-2^{k+1} = 0$$

$$n = 2^0+2^1+\cdots+2^{k+1} = \underbrace{\sum_{i=0}^{k+1} 2^i}_{\text{Geometric Sum}} = 2^{k+2}-1$$

$$\log_2(n+1) = \log_2 2^{k+2} = k+2$$

$$k = \log_2(n+1) - 2$$

$$f(n) = \underbrace{c_2 + c_2 + c_2 + c_2 + c_2 + \cdots + c_2}_{} + \underset{c_1}{f(0)}$$

$$= c_2(k+2) + c_1$$

$$\boxed{f(n) = c_2 + \log_2(n+1) + c_1} \text{ is } O(\log n)$$