

CSC 421/Applied Algorithms and Structures Problem set 2

Name: Kavana Manvi Krishnamurthy
Depaul ID: 2158984

Exercise Search

a) Write down the recurrence relation counting the number of times the **if key = A[m]** comparison is performed.

Recurrence relation is given by the formula:

$$T(n) = r T\left(\frac{n}{c}\right) + f(n)$$

Where,

Number of recursions, $r=2$

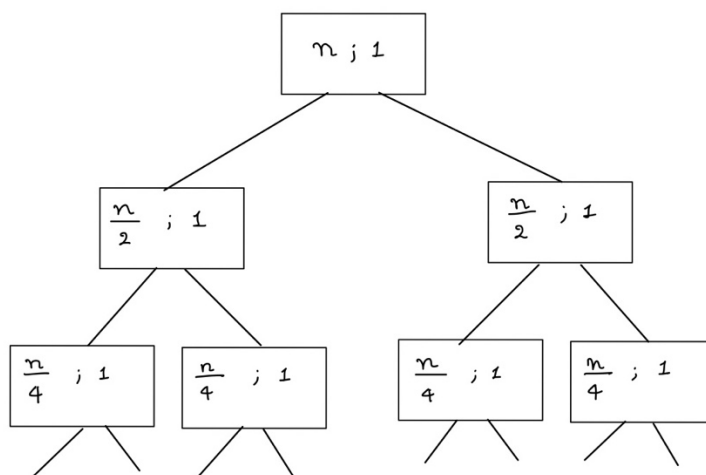
Number of times input is divided, $c=2$

Number of comparison steps, $f(n) = 1$

Substituting the recurrence relation:

$$T(n) = 2 T\left(\frac{n}{2}\right) + 1$$

b) Draw a recursion tree (levels 0, 1, and 2) corresponding to that recurrence relation.



c) Write down the asymptotic formula (that is, using big-Oh notation) derived from that recursion tree.

$$T(n) = O(\log_2 n)$$

At each level of the recursion tree, the size of the problem is halved. Therefore, the height of the recursion tree, which represents the number of comparisons made is $O(\log_2 n)$.

Exercise BuildTree

a) Write down the recurrence relation counting the number of times a comparison is performed inside LINEARSEARCH.

Recurrence relation is given by the formula:

$$T(n) = r T\left(\frac{n}{c}\right) + f(n)$$

Where,

Number of recursions, $r=2$

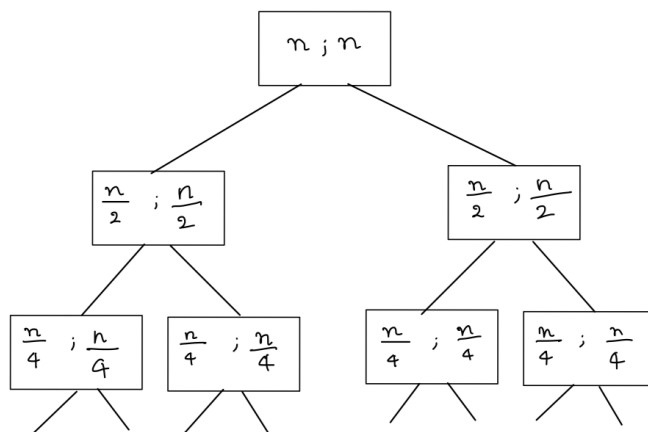
Number of times input is divided, $c = 2$

Number of comparison steps, $f(n) = n$

Substituting the recurrence relation:

$$T(n) = 2 T\left(\frac{n}{2}\right) + n$$

b) Draw a recursion tree (levels 0, 1, and 2) corresponding to that recurrence relation.



c) Write down the asymptotic formula (that is, using big-Oh notation) derived from that recursion tree.

$$T(n) = O(n \log_2 n)$$